MODELING CONTINUOUS IED SUPPLY CHAINS

THESIS

Tony Liu
Second Lieutenant, USAF

AFIT-ENC-14-M-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENC-14-M-02

MODELING CONTINUOUS IED SUPPLY CHAINS

THESIS

Presented to the Faculty

Department of Mathematics and Statistics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Applied Mathematics

Tony Liu, B.A.

Second Lieutenant, USAF

March 2014

AFIT-ENC-14-M-02

MODELING CONTINUOUS IED SUPPLY CHAINS

Tony Liu, B.A.
Second Lieutenant, USAF

Approved:

//signed//	7 March 2014
Kevin R. Pond, Captain, USAF, PhD (Chairman)	Date
//signed//	7 March 2014
Benjamin F. Akers, PhD (Member)	Date
//signed//	7 March 2014
William P. Baker, PhD (Member)	Date

AFIT-ENC-14-M-02

Abstract

Improvised Explosive Devices continue to be a main weapon used by terrorists. The Joint IED Defeat Organization defined three main strategies to fight IEDs: attack the network, defeat the device and equip military forces. This thesis provides models that can give coalition forces a different perspective on how to fight IEDs.

We begin by first developing a model of the supply chain terrorists use to develop, emplace and detonate IEDs. Our model contains four states in which IEDs can exist in: construction (C), emplaced (E), detonated (D) and found by coalition forces (F). We also have five rate parameters representing the flow rates of IEDs from one state to another. Over a given period of time, coalition forces can collect data on the number of IEDs detonated and found. From here, we apply a least squares method to obtain the parameter set for our supply chain model that best fits the collected data. Minimizing our least squares equation allows us to estimate where IEDs are located and how fast they are being moved through the entire supply chain. Using this, we can judge the impact of past efforts. Currently, the only metric coalition forces have for judging counter-IED efforts are the number of IEDs detonated and found. By solving our least squares problem, we can estimate the number of IEDs constructed and emplaced. This offers a more concrete metric for determining how well coalition forces are attacking the network and defeating the device and also offers insight on how to continue fighting IEDs.

Our research focuses on two cases for the behavior of the IED supply chain. In the first case, we assume flow rates are constant. In the second case, we develop a more complicated model in which the rates that IEDs flow from one state to another vary. This model attempts to mimic basic strategies in which the rates are affected by actions from each side. We apply a least squares method on the new parameters to fit our data and to estimate the behavior of the IED supply chain.

Table of Contents

Page

List of Figures

List of Tables

List of Acronyms

Acronym	Definition	
C-IED	Counter-IED	2
COIC	C-IED Operations Integration Center	4
COIN	counter insurgency	7
DoD	Department of Defense	2
IED	Improvised Explosive Device	1
JIEDDO	Joint IED Defeat Organization	2
RK4	4-th order Runge-Kutta scheme	25
TTPs	tactics, techniques and procedures	2

MODELING CONTINUOUS IED SUPPLY CHAINS

I. Introduction

1.1 Background

Improvised Explosive Devices (IEDs) continue to be a main weapon used by terrorists against coalition forces overseas. In fact, IEDs have killed more American soldiers in Afghanistan than anything else since the war began in 2001 [28]. The fact that improvised bombs are effective, unpredictable and cheap to develop makes them the weapon of choice. In the battle against IEDs, coalition forces face a more unpredictable problem. As a result, IEDs pose a major threat to the safety of our deployed troops. Furthermore, IED development is becoming more and more sophisticated. Enemy forces are constantly refining their methods for producing and placing IEDs, using different materials and changing their tactics as the U.S. develops and indoctrinates countermeasure strategies. What results is a cat and mouse battle over the overall affect of IEDs.

This section discuss the measures that U.S. forces have developed in order to counter the production and use of IEDs. We introduce the strategies that policy makers have identified as the best courses of action to mitigate or eliminate the threat of IEDs. We also discuss different units and organizations that have been created for the fight against IEDs as well as how effective they have been in countering IEDs. Lastly, we develop intuition on how this thesis will help in the fight against IEDs.

1.2 Current Efforts and Policies

To address the pressing threat that IEDs pose, the White House published a report in February of 2013 titled *Countering Improvised Explosive Devices* [4]. The report acknowledged that the terrorists and criminals responsible for these attacks are resilient, technologically adept and adaptable. They employ the most recent and successful tactics, techniques and procedures (TTPs) gained from experience in Iraq, Afghanistan and around the world. The use of IEDs worldwide has increased in recent years, with the number of attacks exceeding 4,000 in 2011 [4]. Specifically, this report dives into the policy which the White House believes is the most effective in countering IEDs, noting that the U.S. must not become complacent to ensure national security. By building upon existing policy and strategy, the White House looks to implement measures to counter IEDs by (1) leveraging, integrating and aligning U.S. government efforts, (2) enhancing focus on protecting American lives and (3) promoting cooperation with governmental, international and private sector partners. Furthermore, the report goes on to highlight eight different courses of action which will help put this policy into effect. These courses of action include increasing engagement, exploiting information and materials from attacks, advancing intelligence and information analysis, maintaining deployable Counter-IED (C-IED) resources, screening, detecting and protecting people and resources, safeguarding explosives and precursor materials, coordinating and standardizing equipment and training and enhancing operational planning [4]. The policies of this report underscore the fact that the fight against IEDs continues to be a cat and mouse game and continues to urge the need for U.S. forces to develop and evolve strategies to counter IEDs.

1.2.1 JIEDDO.

In 2006, the Deputy Secretary of Defense ordered the creation of the Joint IED Defeat Organization (JIEDDO). The main purpose of this organization is to lead, advocate and coordinate all of the efforts in defeating IEDs for the Department of Defense (DoD) [29]. Specifically, JIEDDO aims to identify, fund and assess initiatives that provide IED countermeasure solutions. The organization is primarily supported by the U.S. Army.

Throughout the research led by JIEDDO, the DoD has come to solidify their primary strategy for battling IEDs. The three areas of focus are: (1) attack the network, (2) defeat the device itself once emplaced and (3) equip military forces with C-IED techniques [29].

1.2.1.1 Attack the Network.

The first area of focus on defeating IEDs, attack the network, is aimed at performing offensive operations against the complex network of infrastructure that supports the production IEDs [14]. Specifically, this focal point was envisioned to improve our forces' success when attempting to disrupt the enemy's ability to fund, develop and emplace IEDs. The main efforts in attacking the network include: C-IED intelligence, weapons technical intelligence, persistent surveillance, reconnaissance, informations operations, counter-bomber targeting, IED technical and forensic exploitation and disposal of unexploded and captured ordnance [14].

It is important to recognize that enemy forces uses extensive local and international supply chains in order to meet the constant need to create IEDs. Furthermore, these organizations may use informal money transfers and other localized means to acquire the materials needed to produce IEDs. Some enemies produce IEDs in the form of homemade explosives using widely available chemicals while others operate through providers with a more traditional, operating supply chain. This creates the need for intelligence to trace and highlight the many links of enemy IED supply chains so that coalition forces can quickly and effectively interdict suppliers [3].

Since 2007, JIEDDO has delivered over 14,000 intelligence products, trained over 25,000 personnel and deployed over 1,000 personnel in order to support efforts in attacking the network. JIEDDO further breaks down the first area of focus into three tactical areas: (1) gain valuable intelligence, (2) build relationships and (3) neutralize the enemy [14].

To support the efforts in attacking the network, the C-IED Operations Integration Center (COIC) was established in 2007. COIC's network includes over 30 intelligence agencies and other federal agencies that support the effort. COIC provides ease of access to many databases, giving U.S. forces a vast supply of data and information needed to attack the network. Since 2007, COIC's pattern analysis team increased future accident prediction by 85 percent. COIC also led to the discovery and capture of 497,000 pounds of explosive material, which equates to over 12,000 IEDs [14]. In 2008, a forensic exploitation team was deployed to Baghdad to provide enhanced weapons intelligence, forensic exploitation and information fusion capability. Consequently, it helped end improvised rocket-assisted mortar attacks [29]. In building strong relationships and supporting critical organizations, the U.S. and coalition forces can continue to illuminate IED supply chains and reveal the nature of IED networks.

1.2.1.2 Defeat the Device.

The second area of focus includes initiatives that help detect and disarm IEDs that have already been emplaced in order to reduce the deadly effects of IED detonations. This area of focus has led to the implementation of systems that can identify suspicious solids and liquids as well as vehicle mounted IED detection systems and IED detection robots. Several programs were developed as a result of the need to defeat IEDs. One such program includes IED detection kits that can find both metallic and non metallic devices. Further examples of defeating IEDs include Task Force ODIN (Observe, Detect, Identify and Neutralize) which was deployed in 2006. The unit used aviation methods in order to maintain persistent surveillance over areas of high IED risk. This was accomplished using unmanned aerial vehicles [15].

1.2.1.3 Equip Military Forces.

The last area of focus entails equipping military forces with proper C-IED techniques. This includes pre-deployment C-IED training which emphasizes the understanding of current enemy IED capabilities and TTPs. To accomplish this, the Joint Center of Excellence in Ft. Irwin, Ca began running training programs which emphasized more realistic simulations. Other training enhancements include the use of small village complexes, identifying and reporting homemade explosives, improved simulation training, home station training and the development of C-IED mobile assistance training teams. Many organizations are in support of this third area of focus. The U.S Marine Corps Training and Education Command, U.S. Joint Forces Command, U.S. Army Forces Command and the combat training centers have all contributed to increasing training standards and raising IED awareness [16].

1.2.1.4 JIEDDO 2012-2016 Strategy Plan.

JIEDDO also released its 2012-2016 strategy in 2012 [13]. Even with dwindling operations overseas, JIEDDO foresees continuously evolving bomb threats. In fact, JIEDDO predicts that IEDs will continue to be used by members of the threat continuum. This threat continuum contains actors such as smugglers, criminals, pirates, drug traffickers and terrorists. Moreover, JIEDDO has pointed out the spread of IED use to places such as Thailand and Norway. U.S. reports have also concluded that from January to November of 2011, a total of 6,832 bombing incidents across 111 nations killed 12,286 people. With this, comes a warning from JIEDDO that IEDs will continue to make their way onto U.S. soil [13].

The execution of U.S. policies regarding C-IED measures has led to successes overseas. Reports confirm that in 2012, there was a 46 percent decline in deaths and a 50 percent decline in wounds resulting from IEDs in Afghanistan. The decline was linked to an increase in surveillance equipment, metal detectors and intensive training. Each of these supported one of the three areas of focus outlined by JIEDDO. Although the effect of IEDs on U.S. troops was heavily mitigated, Afghan troops suffered a 124 percent increase in the number of IED attacks against them. To address this pressing issue, U.S. advisers have helped train two dozen Afghan units in C-IED measures [5] [28].

1.2.2 Research Objectives.

In the fight against IEDs, JIEDDO has laid out three main focal points for the U.S. to aim their efforts: attack the network, defeat the IED and equip military personnel. Our research intends to give U.S. forces a better perspective on fighting IEDs. We begin by first developing a model of the supply chain enemy forces use to develop, emplace and detonate IEDs. From here, using the research methods discussed in the methodology section, we apply a least squares method to estimate where the IEDs are located as well as how fast they are being moved throughout the supply chain. Using this, we can judge the impact of our past efforts. Solving a least squares problems provides an estimate for the number of IEDs constructed and emplaced, information we did not have access to, to start with. Without this information, coalition forces would have to use the number of IEDs detonated and found to measure how well they are they are attacking the network(interdicting IEDs before being emplaced) and defeating the device(interdicting emplaced IEDs). Using our estimate for IEDs constructed and emplaced, we can judge the effectiveness of our efforts in attacking the network and defeating the device directly. This estimate can also provide insight on how to continue to fight IEDs in the future.

II. Literature Review

2.1 Introduction

On top of the research JIEDDO has performed to fight IEDs, there have been numerous academic research papers published for the same purpose. In this section, we mention some of the previous work aimed at countering IEDs and how they have contributed to the battle against IEDs. Furthermore, we also take a look at supply chain model examples and how they can be used to develop our model for the IED supply chain process.

2.2 Previous C-IED Research

2.2.1 Modeling Insurgency, Counter-Insurgency and Coalition Strategies and Operations .

In 2012, David and Kristin Arney published *Modeling Insurgency, Counter-insurgency, and Coalition Strategies and Operations* [7]. In this report, David and Kristin Arney analyzed insurgency and counter insurgency (COIN) operations using a large scale system of differential equations to model a dynamically changing coalition network. Using this model, the two analyzed how leadership, promotion, recruitment, financial resources, operational techniques, network communications, coalition cooperation, logistics, security, intelligence, infrastructure development, humanitarian aid, and psychological warfare affected COIN tactics, operations and strategy. A total of 19 dependent factors, with 19 equations and over 80 parameters were developed for the model. Using these network models and the system of differential equations, Arney and Arney then ran an eight-stage test scenario to demonstrate the behavior or their model. Using this model, the two were able to determine optimal counter-insurgency strategies [7]. Similar to how Arney and Arney analyzed how certain parameters affected their insurgency, counter-insurgency model, we wish to analyze how certain parameters will affect our IED supply chain model. In this thesis, we will also employ the use of systems of differential equations and parameters to study the behavior of our model. Specifically, we model the IED supply chain and search our parameter set for which parameter values best fit collected IED data.

2.2.2 JIEDDO Proposal Selection Model.

In 2009, 1Lt Christina Willy authored a master's thesis titled *Robust Sensitivity Analysis for the JIEDDO Proposal Selection Model* [32]. In this thesis, Willy analyzes JIEDDO's proposal solicitation process; a tool that allows organizations both military and civilian to request funding for the development of specific C-IED projects. Willy employed decision analysis methods to the JIEDDO proposal selection process in order to help decision makers filter out proposals that did not adequately satisfy C-IED objectives. Specifically, Willy applied value focused thinking to analyze proposals [32]. Using this technique, Willy evaluates JIEDDO's pre-developed proposal value model to determine whether or not the value model accurately reflected JIEDDO's decision process. Willy's research methods were aimed at gauging the efficacy of C-IED projects before being implemented in order to determine which projects to select. In this thesis, we wish to provide intuition on how effective certain projects are after being implemented. After applying our least squares method, we will obtain an estimate for the behavior of the entire supply chain. With this knowledge, we can gauge how effective our past C-IED efforts were by looking at its effect on our IED supply chain at past times.

2.2.3 Agent Based Simulation.

Other works involved with C-IED includes Dr. Anthony Dekker's *Agent-Based Simulation for Counter-IED: A Simulation Science Survey* [10]. This paper uses agent-based simulations to model the human networks that place IEDs. Dekker categorizes two main human aspects of IED placement. The first is motivation which affects the number of IEDs placed per day and the second is cognition which affects the choice of location for IEDs. Dekker goes on to label counter-insurgency forces as blue forces and insurgency forces as red forces. He advocates simulations in which he models motivation as being affected by attitudes and emotions within society and models cognition as being affected by learning and adapting to opposing forces' responses. Dekker terms the co-evolutionary aspect of COIN operations: blue actions alter how red forces conduct operations and vice versa [10]. He then goes on to mention that these simulations can be used to explore this adaptive behavior. Dekker's research aims to create a model of how IED placement strategies are affected by the actions of other players. In this thesis, we will model our IED flow rate parameters to also be affected by the actions of other players. In our nonlinear model (see Section 3.1.3), we will set up a model where the flow of IEDs from one state to another is affected by both coalition and enemy actions.

2.2.4 Optimization, Games and Adaption.

Dekker also published *Optimization, Games, Adaptation: Three Perspectives on Operations Research for Counter-IED* [11]. In this paper, Dekker examines optimization approaches for C-IED actions. Dekker also analyzes game theoretic approaches where both blue and red forces can adapt to each others' actions. In terms of optimization, Dekker looks at which blue (C-IED) strategy ensures the best chance of survival when having to traverse a simulated terrain grid. This approach assumes a fixed red strategy. He then determines optimal strategies for both sides. Dekker then goes even further to incorporate adaptivity, in which enemies can devise counter counter-IED strategies. He concludes that the most rapidly adapting side has the advantage [11].

2.3 IED Supply Chain Models

JIEDDO's first area of focus in their C-IED policy stresses the need for the U.S. to attack the networks used by enemy forces. In order to better understand how enemy forces are employing IEDs, we must first model the process of producing and emplacing IEDs through a supply chain perspective. By modeling the process in which enemy forces obtain material and construct, emplace and detonate IEDs, we can study the behavior of IED threats as we aim to implement C-IED strategies. Specifically, we will represent the lifespan of an IED using a supply chain model.

Although enemy forces may have non-traditional avenues for obtaining materials needed to rapidly construct IEDs, the general supply chain process for IEDs remain similar to those of traditional products. The traditional supply chain is pictured in Figure 2.1 taken from Beamon's *Supply Chain Design and Analysis: Models and Methods* [8]. For the case of IEDs, there are both traditional and non-traditional suppliers for materials, represented by a single supplier entity in the figure below. Manufacturing facilities may be paralleled by factories where enemies construct IEDs or even homes where homemade IEDs are built. Further in the process, IEDs may be stored or transported straight to the battlefield where they are then emplaced and left to detonate.

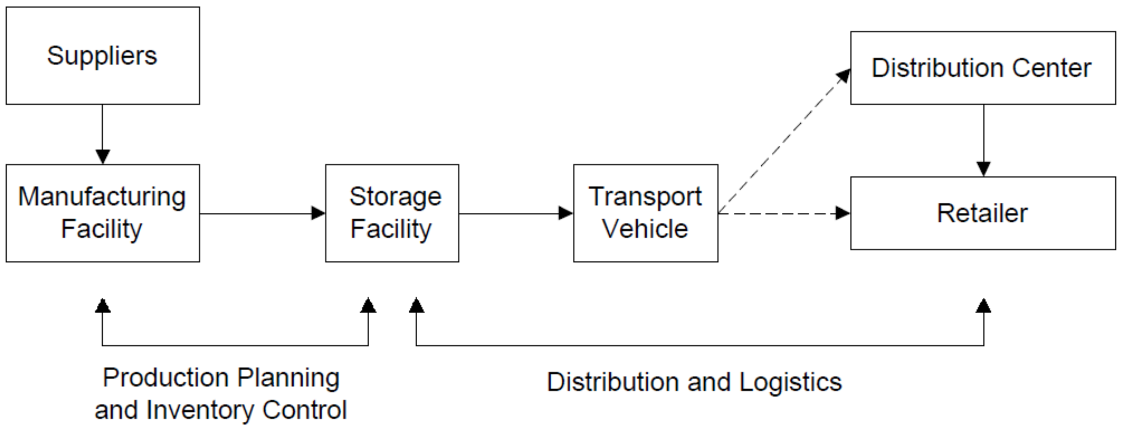


Figure 2.1: Supply Chain Diagram

Currently, there are published reports that demonstrate supply chain models directly applied to IED life spans. Specifically, *Modeling Behavioral Activities Related to Deploying IEDs in Iraq* by Weiss, Whitaker, Briscoe and Trewhitt presents a model which takes into account an IED disruption model, materials and supplies gathering model and an IED motivation model [9]. The collective model is depicted in Figure 2.2 below.

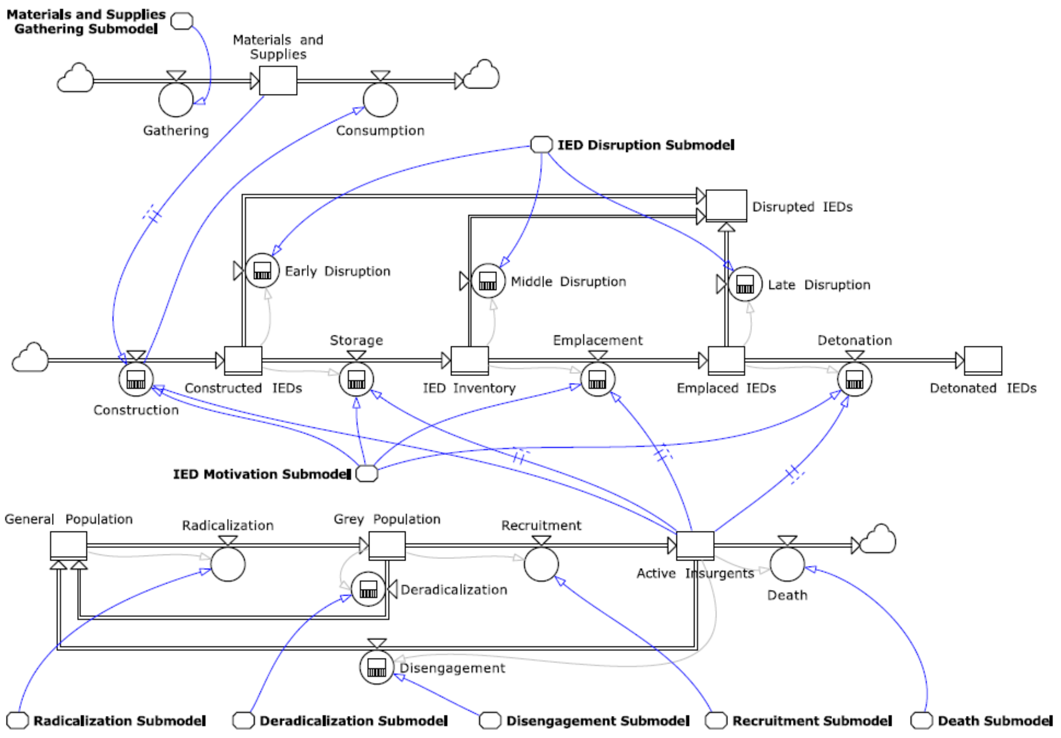


Figure 2.2: Diagram of IED Core Model

Looking at just the IED disruption submodel, we have a supply chain model that illustrates the life cycle of IEDs as they go from being constructed to being detonated. This model introduces five states in which IEDs can exist: construction, inventory storage, emplaced, detonated and disrupted. According to the submodel, U.S. forces can disrupt the detonation of IEDs in three spots: early disruption which corresponds to seizing IEDs while they are being constructed, middle disruption which corresponds to seizing IEDs when they are in storage and late disruption which corresponds to neutralizing emplaced IEDs [9].

Weiss, Whitaker, Briscoe and Trewhitt go on to use their models for what-if analysis. Using these techniques they illustrate the factors and influences that are more significant than others. An example they present is using the model to determine the effects of various deterrents on the overall rate of IED detonation. The factors that were considered in this case were insurgent disengagement effectiveness, IED disruption effectiveness and supply gathering interference. From here, studies then determined that disengagement yielded the largest long-term (36 months) reduction in overall IED detonations while increase in supply interference yielded the smallest [9]. The models that Weiss, Whitaker, Broscoe and Trewhitt used in their technical report regarding IEDs in Iraq is used to formulate our own IED supply chain model which is discussed in the next section.

III. Methodology

3.1 Modeling the Supply Chain

3.1.1 Mathematical Model Background.

Using the work of Weiss, Whitaker, Broscoe and Trewhitt [9], we develop our own supply chain model for IED production. We focus on only the IED disruption submodel from Figure 2.2, using four stages to model the IED supply chain: construction, emplaced, detonated and found. We also incorporate disruption possibilities similar to the IED disruption submodel. In our model, we include two opportunities for disruption, one during the transition from construction to being emplaced and one from being emplaced to detonation. Our IED supply chain model is depicted in Figure 3.1 below.

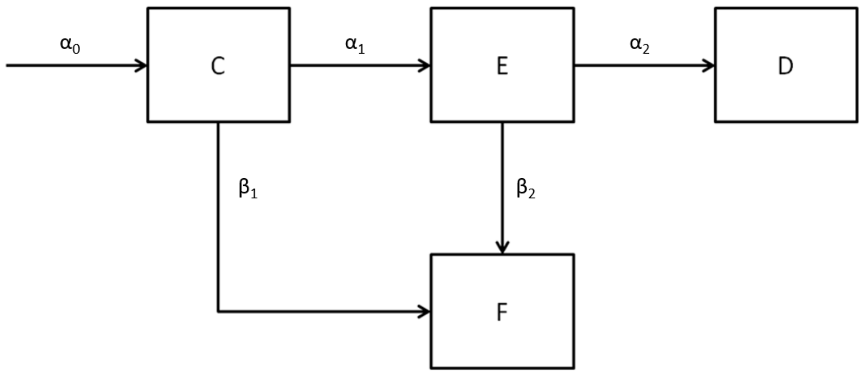


Figure 3.1: IED Supply Chain Model

In our IED model above, the boxes represent the four states of the IED lifespan: *C* for construction, *E* for emplaced, *D* for detonated and *F* for found. The arrows represent transitions while α_i and β_i represent the transition rates. We see that α_1 and α_2 correspond to the rate at which our enemies are transporting IEDs from the construction state to the emplaced state and to the rate at which emplaced IEDs are being detonated, respectively. Also, α_0 corresponds to the rate at which IED constructors are receiving materials to produce IEDs. Lastly, β_1 and β_2 represent the rates in which we disrupt IEDs from the constructed and emplaced states, respectively.

It is important to note one key difference between our supply chain model and the one used by Weiss, Whitaker, Broscoe and Trewhitt [9]. Instead of using five states, we merge the inventory storage state into the construction state to create a more compact four state model. Similarly, we combine early and middle disruption of the IED disruption submodel into one disruption rate β_1. The reason we chose to make these simplifications is to adhere to JIEDDO's three main focal points in battling IEDs. We see that the first focal point which aims to disrupt the enemy's ability to fund, develop and emplace IEDs corresponds to both early and middle disruption while the second focal point, which aims to disarm IEDs that have already been emplaced corresponds to late disruption.

Using this supply chain model to depict how enemies produce, emplace and detonate IEDs we then develop a mathematical model. In developing this model for the IED lifespan, we aim to mimic existing models that have been used to represent behavior similar to that of the IED supply chain. One such model is the Susceptible, Infected and Removed (SIRS) model used in epidemiology [30].

The SIRS model is an important part of mathematical biology. It is an important tool used for gauging the impact of different vaccination programs on the control and eradication of certain diseases. In this model, there exist three states, each representing a portion of the population. The susceptible group contains those who can receive the virus from the members of the infected group. The last group is the removed group which represent those who have recovered from the disease. They are then free of the infection but rejoin susceptible group in the SIRS model. Figure 3.2 below illustrates the SIRS model [30].

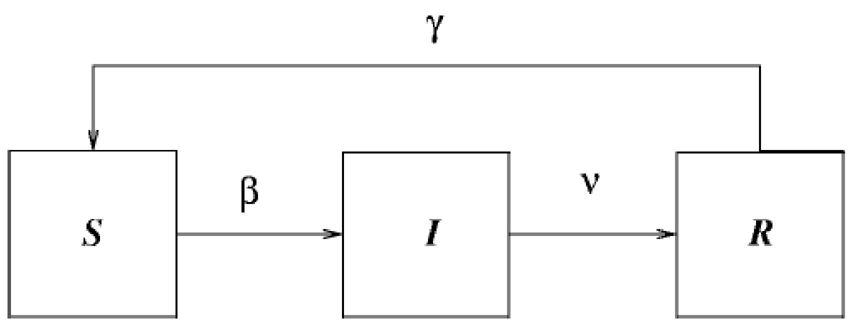


Figure 3.2: SIRS Model

In this model, we see that β represents the rate at which susceptible victims become infected, ν represents the rate at which infected members become removed and γ represents the rate at which removed members rejoin the susceptible population. Mathematically modelling this dynamic process, we get the following equations for the population.

$$\dot{S} = -\beta S I + \gamma R \tag{3.1}$$

$$\dot{I} = \beta S I - \nu I \tag{3.2}$$

$$\dot{R} = \nu I - \gamma R \tag{3.3}$$

The SIRS model is widely used in mathematical biology and provides us a way to mathematically model our IED supply chain. Both the SIRS model and our IED supply chain model possess similar behavior; there exists states which individuals can exist in and there exists flow rates in which individuals move from one state to another. Thus, it is reasonable to model our supply chain to be similar to the SIRS model. More information about the SIRS model can be found in [30].

Similarly, we derive equations to model the dynamic process of our IED supply chain model. For this thesis, we consider two cases. In the first case, we assume that our rates, α_i and β_i, are constant parameters. In a later section we will introduce time dependent rates.

3.1.2 Formulating the Linear Model.

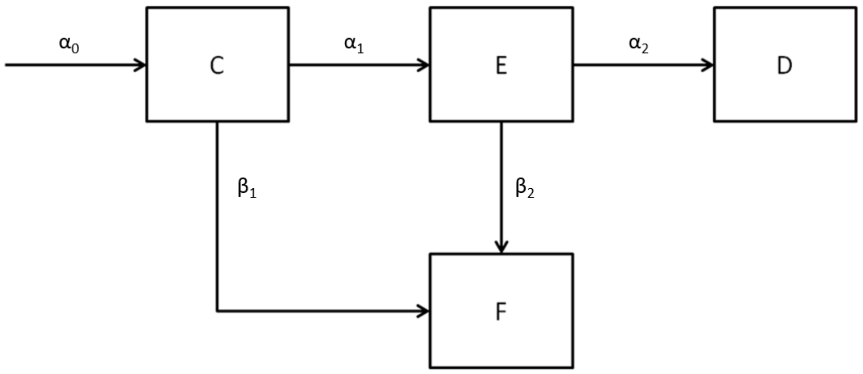


Figure 3.3: IED Supply Chain Model

Assuming constant rates and applying the same methods used to derive the SIRS model's differential equations, we derive the following system of equations for our IED model:

$$\dot{C} = \alpha_0 - (\alpha_1 + \beta_1)C \tag{3.4}$$

$$\dot{E} = \alpha_1 C - (\alpha_2 + \beta_2)E \tag{3.5}$$

$$\dot{D} = \alpha_2 E \tag{3.6}$$

$$\dot{F} = \beta_1 C + \beta_2 E \tag{3.7}$$

Here it is important to consider the units in this system. Since C, E, D and F represent our states, they must have units of IEDs, thus $\dot{C}$, $\dot{E}$, $\dot{D}$ and $\dot{F}$ have units of $\frac{IEDs}{time}$. α_0 also has units of $\frac{IEDs}{time}$, while α_1, $\alpha_2, \beta_1, \beta_2$ (and β, γ, ν from the SIRS model) have units of $\frac{1}{time}$.

Physically, this means that α_0 represents a flow rate: the number of IEDs flowing into the construction phase over time while the other four parameters represent frequency. That is, α_1 represents the amount of IEDs in the construction state that flow to the emplaced state over a given time period. α_2, β_1 and β_2 behave similarly, thus we see that α_0 has different units compared to the rest of the rate parameters

We now have a system of four equations for our IED model. If α_i and β_i are constants these equations are all linear and have constant coefficients, allowing us to solve $C(t)$, $D(t)$, $E(t)$ and $F(t)$ explicitly. Solving each of these differential equations, we get the following solutions:

$$C(t) = \frac{\alpha_0}{r_1} + k_1 e^{-r_1 t} \tag{3.8}$$

$$E(t) = \left(\frac{\alpha_0\alpha_1}{r_1 r_2}\right) - \frac{\alpha_1 k_1 e^{-r_1 t}}{r_1 - r_2} + k_2 e^{-r_2 t} \tag{3.9}$$

$$D(t) = \left(\frac{\alpha_2\alpha_0\alpha_1}{r_1 r_2}\right) t + \frac{\alpha_2\alpha_1 k_1 e^{-r_1 t}}{r_1 (r_1 - r_2)} - \frac{\alpha_2 k_2 e^{-r_2 t}}{r_2} + k_3 \tag{3.10}$$

$$F(t) = \left[\frac{\beta_1\alpha_0}{r_1} + \left(\frac{\beta_2\alpha_0\alpha_1}{r_1 r_2}\right)\right] t - \frac{\beta_1 k_1 e^{-r_1 t}}{r_1} + \frac{\beta_2\alpha_1 k_1 e^{-r_1 t}}{r_1 (r_1 - r_2)} - \frac{\beta_2 k_2 e^{-r_2 t}}{r_2} + k_4 \tag{3.11}$$

Where k_i are integration constants that are solved for by imposing initial conditions and r_i=α_i+β_i. In this thesis, we consider k_i parameters as well. We do not know how many IEDs are in C and D at time $t = 0$, thus k_1 and k_2 will need to be estimated using our least squares problem. Although we do know how many IEDs are detonated and found at time $t = 0$, we still allow them to be free parameters. Doing so allows us to better fit data in the least squares problem defined in section 3.2.2. Solving for initial conditions, we then get the following equations:

$$C(0) = \frac{\alpha_0}{r_1} + k_1 \tag{3.12}$$

$$E(0) = \left(\frac{\alpha_0\alpha_1}{r_1 r_2}\right) - \frac{\alpha_1 k_1}{r_1 - r_2} + k_2 \tag{3.13}$$

$$D(0) = \frac{\alpha_2\alpha_1 k_1}{r_1\left(r_1 - r_2\right)} - \frac{\alpha_2 k_2}{r_2} + k_3 \tag{3.14}$$

$$F(0) = -\frac{\beta_1 k_1}{r_1} + \frac{\beta_2\alpha_1 k_1}{r_1\left(r_1 - r_2\right)} - \frac{\beta_2 k_2}{r_2} + k_4 \tag{3.15}$$

3.1.3 Formulating the Nonlinear Model.

We now formulate the supply chain model which mimics a basic strategy in which both coalition forces and insurgents adapt. We can model this situation by assuming our flow rates are functions of $C(t), D(t), E(t)$ and $F(t)$. Let us consider each of the new flow rates, denoted by a tilde.

3.1.3.1 Emplaced to Detonated: $\tilde{\alpha_2}$.

In our IED supply chain model, $\tilde{\alpha_2}$ physically represents the rate at which emplaced IEDs detonate. For our IED model, we assume the rate at which IEDs detonate is independent of coalition or enemy forces' actions. That is, we assume that neither coalition forces nor insurgents have the power to influence the amount of emplaced IEDs that detonate and that this rate is independent of both sides' actions. Thus, in this case, we keep $\tilde{\alpha_2}$=α_2 as constant.

3.1.3.2 Found from Construction: $\tilde{\beta_1}$.

$\tilde{\beta_1}$ represents the rate at which coalition forces find IEDs in the construction state. In a basic strategy, we expect that the more IEDs that detonate, the more U.S. forces will focus on finding IEDs in construction. That is, $\tilde{\beta_1}$ must be directly influenced by the rate at which D grows, $\dot{D} = \alpha_2 E$. Thus, we suggest the rate model:

$$\tilde{\beta_1} = \beta_1 + \gamma_1\alpha_2 E$$

where β_1 is a constant that represents a minimum flow rate in case $\dot{D} = 0$ and γ_1 is also a constant representing the influence of $\dot{D}$ on the coalition force's strategy.

3.1.3.3 Found from Emplaced: $\tilde{\beta}_2$.

We assume $\tilde{\beta_2}$ to have similar dependence as $\tilde{\beta_1}$. That is, if more IEDs are detonating, coalition forces will look to find more emplaced IEDs. Thus, we suggest the rate model:

$$\tilde{\beta_2} = \beta_2 + \gamma_2\alpha_2 E$$

where β_2 is a constant that represents a minimum flow rate in case $\dot{D} = 0$ and γ_2 is also a constant representing the influence of $\dot{D}$ on the coalition force's strategy. We see that γ_1 and γ_2 essentially represents how coalition forces decide to divide their efforts between attacking the network and defeating the device as a result of IED detonations.

3.1.3.4 Construction to Emplaced: $\tilde{\alpha}_1$.

We assume that the enemy force's basic strategy will be to emplace more IEDs if more are detonating and/or are being found after being emplaced, represented by $\dot{D} = \alpha_2 E$ and $\tilde{\boldsymbol{\beta}}_2 E = (\beta_2 + \gamma_2\alpha_2 E)E$ respectively. Thus, we suggest the rate model:

$$\tilde{\alpha_1} = \alpha_1 + \gamma_3(\beta_2 + \gamma_2\alpha_2 E + \alpha_2)E$$

where α_1 is a constant that represents a minimum flow rate and γ_3 is also a constant representing the influence of found and detonated IEDs on the insurgent force's strategy.

3.1.3.5 Raw Material to Construction: $\tilde{\alpha}_0$.

We wish to impose a limit for $\tilde{\alpha_0}$ as a function of $C(t)$. That is, as $C(t)$ reaches some critical capacity, C^*, $\tilde{\alpha_0}$ must approach 0 and when $C(t)$ approaches 0, $\tilde{\alpha_0}$ must approach a maximum flow rate. Physically, this represents the speeding up and slowing down of inflow of resources as the number of IEDs in construction reaches 0 and its maximum capacity, respectively. Thus, we suggest the rate model:

$$\tilde{\alpha_0}=(1 - \tfrac{C}{C^*})\alpha_0$$

where α_0 is the maximum flow rate. We see this formula for $\tilde{\alpha_0}$ meets our criteria outlined above.

We note that this is not the only function that can satisfy our criteria. We impose:

$$\tilde{\alpha}_0 = f(C^*, C^*)\alpha_0$$

$$f(C^*, C^*) = 0$$

$$f(0, C^*) = 1$$

This can be acheived by other functions, for example:

$$f(C, C^*) = \left(1 - \frac{C}{C^*}\right)^p \qquad p \in \mathbb{R}$$

For our model, we choose $p = 1$.

Now, we see our IED supply chain is modeled by the system of differential equations in 3.16-3.19. Furthermore, we notice that our nonlinear model contains our linear model. That is, if we let C^* approach infinity and set $\gamma_1 = \gamma_2 = \gamma_3 = 0$, we have our linear model, however, our nonlinear case includes an extra four degrees of freedom in C^*, γ_1, γ_2 and γ_3. Thus, the goal is to apply least squares methods on the nonlinear case to attempt to improve our results from the linear case.

$$\dot{C} = (1 - \frac{C}{C^*})\alpha_0 - (\alpha_1 + \gamma_3(\beta_2 + \gamma_2\alpha_2 E + \alpha_2)E + \beta_1 + \gamma_1\alpha_2 E)C \tag{3.16}$$

$$\dot{E} = (\alpha_1 + \gamma_3(\beta_2 + \gamma_2\alpha_2 E + \alpha_2)E)C - (\alpha_2 + \beta_2 + \gamma_2\alpha_2 E)E \tag{3.17}$$

$$\dot{D} = \alpha_2 E \tag{3.18}$$

$$\dot{F} = (\beta_1 + \gamma_1\alpha_2 E)C + (\beta_2 + \gamma_2\alpha_2 E)E \tag{3.19}$$

$$< C(0), E(0), D(0), F(0) > \quad = \quad < C_0, E_0, D_0, F_0 > \tag{3.20}$$

3.1.4 Proving Uniqueness of the Solution to the Nonlinear Case.

Before solving our system of differential equations for the nonlinear case, we must first prove that the solution is unique. To do this, we first introduce preliminary definitions and lemmas.

Definition 3.1. A vector function $\boldsymbol{f}(\boldsymbol{x})$ is a Lipschitz continuous function in $\boldsymbol{x}$ if there exists a constant m, the Lipschitz constant, such that

$$||\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{y})|| \leq m||\boldsymbol{x} - \boldsymbol{y}||$$

Lemma 3.2. If $\boldsymbol{f}(\boldsymbol{x})$ has bounded first-partial derivatives in each of x_1,x_2,...,x_n, then $\boldsymbol{f}(\boldsymbol{x})$ is a Lipschitz continuous function [31].

Lemma 3.3. Suppose $\boldsymbol{f(x)}$ is Lipschitz continuous for all $\boldsymbol{x}$, $\boldsymbol{y}$ in some interval $[\boldsymbol{x_0} - \delta, \boldsymbol{x_0} + \delta]$. Then a unique solution $\boldsymbol{x(t)}$ of

$$\boldsymbol{x}'(t) = \boldsymbol{f}(\boldsymbol{x})$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0$$

exists for all t in some interval $t \in (t_0 - \epsilon, t_0 + \epsilon)$ [1].

We assume that α_i, β_i and γ_i are finite and that $C^* \geq 1$, These assumptions stem from our physical representation of the IED supply chain since we do not investigate infinite supply chains or supply chains with construction capacity less than one IED. We also see that $\alpha_i, \beta_i, \gamma_i, C(t), E(t), D(t), F(t) \geq 0$

Theorem 3.4. *Assuming α_i, β_i and γ_i are finite and that $C^* \geq 1$, then the solution to the system of nonlinear differential equations (3.16-3.20) is unique.*

Proof. We see that our system of differential equations can be written in the form

$$\boldsymbol{x}'(t) = \boldsymbol{f}(\boldsymbol{x})$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0$$

where

$$\boldsymbol{x}(t) = \begin{pmatrix} C(t) \\ E(t) \\ D(t) \\ F(t) \end{pmatrix}$$

and $\boldsymbol{x_0}$ is a vector with our initial data for $C(t)$, $E(t)$, $D(t)$ and $F(t)$ as elements. In this case, $\boldsymbol{f}(\boldsymbol{x})$ is a vector function with the right hand sides of equations (3.16 – 3.19) as entries. We now calculate the first partial derivatives of $\boldsymbol{f}(\boldsymbol{x})$:

$$\frac{\partial \boldsymbol{f}}{\partial C} = \begin{pmatrix} -\frac{\alpha_0}{C^*} - (\alpha_1 + \gamma_3(\beta_2 + \gamma_2\alpha_2 E + \alpha_2)E + \beta_1 + \gamma_1\alpha_2 E) \\ \alpha_1 + \gamma_3(\beta_2 + \gamma_2\alpha_2 E + \alpha_2)E \\ 0 \\ \beta_1 + \gamma_1\alpha_2 E \end{pmatrix}$$

$$\frac{\partial \boldsymbol{f}}{\partial E} = \begin{pmatrix} -\gamma_3\beta_2 C - 2\gamma_2\gamma_3\alpha_2 CE - \gamma_3\alpha_2 C + \gamma_1\alpha_2 C \\ \gamma_3\beta_2 C + 2\gamma_2\gamma_3\alpha_2 CE + \gamma_3\alpha_2 C - \alpha_2 - \beta_2 - 2\gamma_2\alpha_2 E \\ \alpha_2 \\ \gamma_1\alpha_2 C + \beta_2 + 2\gamma_2\alpha_2 E \end{pmatrix}$$

$$\frac{\partial \boldsymbol{f}}{\partial D} = 0$$

$$\frac{\partial \boldsymbol{f}}{\partial F} = 0$$

Since each equation in our system is continuous with respect to $C(t)$ and $E(t)$ we see that $C(t)$ and $E(t)$ are bounded over a finite time interval. Let $C(t) \leq \hat{C}$, $E(t) \leq \hat{E}$, $|\alpha_i| \leq \hat{\alpha}_i$, $|\beta_i| \leq \hat{\beta}_i$, $|\gamma_i| \leq \hat{\gamma}_i$ be the bounds. Then we have:

$$\begin{aligned} &|-\tfrac{\alpha_0}{C^*} - (\alpha_1 + \gamma_3(\beta_2 + \gamma_2\alpha_2 E + \alpha_2)E + \beta_1 + \gamma_1\alpha_2 E| \\ &\qquad \leq \hat{\alpha}_0 + (\hat{\alpha}_1 + \hat{\gamma}_3(\hat{\beta}_2 + \hat{\gamma}_2\hat{\alpha}_2\hat{E} + \hat{\alpha}_2)\hat{E} + \hat{\beta}_1 + \hat{\gamma}_1\hat{\alpha}_2\hat{E}) \\ &|\alpha_1 + \gamma_3(\beta_2 + \gamma_2\alpha_2 E + \alpha_2)E| \\ &\qquad \leq \hat{\alpha}_1 + \hat{\gamma}_3(\hat{\beta}_2 + \hat{\gamma}_2\hat{\alpha}_2\hat{E} + \hat{\alpha}_2)\hat{E} \\ &|\beta_1 + \gamma_1\alpha_2 E| \\ &\qquad \leq \hat{\beta}_1 + \hat{\gamma}_1\hat{\alpha}_2\hat{E} \\ &|-\gamma_3\beta_2 C - 2\gamma_2\gamma_3\alpha_2 CE - \gamma_3\alpha_2 C + \gamma_1\alpha_2 C| \end{aligned}$$

$$\leq \hat{\gamma}_3\hat{\beta}_2\hat{C} + 2\hat{\gamma}_2\hat{\gamma}_3\hat{\alpha}_2\hat{C}\hat{E} + \hat{\gamma}_3\hat{\alpha}_2\hat{C} + \hat{\gamma}_1\hat{\alpha}_2\hat{C}$$

$$|\gamma_3\beta_2 C + 2\gamma_2\gamma_3\alpha_2 CE + \gamma_3\alpha_2 C - \alpha_2 - \beta_2 - 2\gamma_2\alpha_2 E|$$

$$\leq \hat{\gamma}_3\hat{\beta}_2\hat{C} + 2\hat{\gamma}_2\hat{\gamma}_3\hat{\alpha}_2\hat{C}\hat{E} + \hat{\gamma}_3\hat{\alpha}_2\hat{C} + \hat{\alpha}_2 + \hat{\beta}_2 + 2\hat{\gamma}_2\hat{\alpha}_2\hat{E}$$

$$|\alpha_2|$$

$$\leq \hat{\alpha}_2$$

$$|\gamma_1\alpha_2 C + \beta_2 + 2\gamma_2\alpha_2 E|$$

$$\leq \hat{\gamma}_1\hat{\alpha}_2\hat{C} + \hat{\beta}_2 + 2\hat{\gamma}_2\hat{\alpha}_2\hat{E}$$

We then consider the infinity-norm: $||A||_\infty = \max_i(\sum_{j=1}^{n} |a_{i,j}|)$ Thus, with our assumptions on α_i, β_i, γ_i and C^* and applying Lemma 3.2 and Lemma 3.3, we have that our solution is in fact unique. Here we note the importance of assuming that $C^* \geq 1$ as it bounds C^* away from 0, thus preventing $\frac{\alpha_0}{C^*}$ from becoming unbounded.

$\square$

For more information regarding the existence and uniqueness of solutions to nonlinear differential equations, refer to [31].

3.1.5 Solving the System of Differential Equations.

In order to solve for $C(t)$, $E(t)$, $D(t)$ and $F(t)$, governed by the system of differential equations in 3.16-3.20, we must employ the use of a numerical differential equations solver. One such solver is MATLAB's ODE45 [25]. Solutions of this numerical solver are obtained using the Runge-Kutta(4,5) method explained in the sections below.

3.1.5.1 Runge-Kutta Methods.

The goal of a Runka-Kutta method is to seek solutions, **x**(t), of

$$\boldsymbol{x}'(t) = \boldsymbol{f}(t, \boldsymbol{x}) \tag{3.21}$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \qquad a \leq t \leq b \tag{3.22}$$

on an interval $a \leq t \leq b$, where $\boldsymbol{f}$ is a given vector valued function of $n + 1$ variables and $\boldsymbol{x}_0$ is a given n-dimensional vector. The Runka-Kutta(4,5) method is an explicit single step method.

Definition 3.5. An explicit single step method for numerical solutions of 3.21-3.22 is a method of the form

$$\begin{aligned} \boldsymbol{x}_0 &= \boldsymbol{x}(t_0) \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + h\boldsymbol{\phi}(t_k, \boldsymbol{x_k}, h) \qquad 0 \leq k \leq N-1 \end{aligned}$$

where $\boldsymbol{\phi}$ is a given function, h is the time step used to discretize our time interval $[a, b]$ and $N + 1$ is the number of time points in our discretization. This method allows us to obtain the solution at discretized time points, $\boldsymbol{x_0}$, $\boldsymbol{x_1}$,...$\boldsymbol{x}_N$.

Definition 3.6. In general, Runge Kutta methods are of the following form

$$\begin{aligned} \boldsymbol{x}_0 &= \boldsymbol{x}(t_0) \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + h\boldsymbol{\phi}(t_k, \boldsymbol{x_k}, h) \qquad 0 \leq k \leq N-1 \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{\phi}(t_k, \boldsymbol{x_k}, h) &= \sum_{r=1}^{R} c_r \boldsymbol{K_r} \\ \boldsymbol{K_1} &= \boldsymbol{f}(t, \boldsymbol{x}) \\ \boldsymbol{K_r} &= \boldsymbol{f}(t + a_r h, \mathbf{x} + h \sum_{s=1}^{r-1} b_{rs} \boldsymbol{K_s}) \\ a_r &= \sum_{s=1}^{r-1} b_{rs} \qquad r = 2, 3, ..., R \end{aligned}$$

Specifically, a method of this form is called an R-stage Runge Kutta method.

Example 3.7. Runge Kutta 4th Order

One well known Runge-Kutta scheme is the 4-th order Runge-Kutta scheme (RK4), where the order refers to the truncation error (in this case $O(h^4)$). The RK4 method takes the form:

$$
\begin{aligned}
\boldsymbol{x}_0 &= \boldsymbol{x}(t_0) \\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \frac{h}{6}[\boldsymbol{K_1} + 2\boldsymbol{K_2} + 2\boldsymbol{K_3} + \boldsymbol{K_4}] \\
\boldsymbol{K_1} &= f(t_k, y_k) \\
\boldsymbol{K_2} &= f(t_k + \frac{h}{2}, y_k + \frac{h}{2}\boldsymbol{K_1}) \\
\boldsymbol{K_3} &= f(t_k + \frac{h}{2}, y_k + \frac{h}{2}\boldsymbol{K_2}) \\
\boldsymbol{K_4} &= f(t_k + h, y_k + h\boldsymbol{K_3}) \\
\boldsymbol{\phi}(t_k, \boldsymbol{x_k}, h) &= \frac{h}{6}[\boldsymbol{K_1} + 2\boldsymbol{K_2} + 2\boldsymbol{K_3} + \boldsymbol{K_4}]
\end{aligned}
$$

In this case, we have

r	c_r	a_r	b_{r1}	b_{r2}	b_{r3}
1	$\frac{1}{6}$	0	-	-	-
2	$\frac{2}{6}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0
3	$\frac{2}{6}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0
4	$\frac{1}{6}$	1	0	0	1

Table 3.1: RK4 Coefficients

For readability, Runge-Kutta coefficients are arranged in what is called a Butcher Tableau of the form:

$$
\begin{array}{c|ccccc}
0 & & & & & \\
a_2 & b_{21} & & & & \\
a_3 & b_{31} & b_{32} & & & \\
\vdots & \vdots & \ddots & & & \\
a_R & b_{R1} & b_{R2} & \cdots & b_{R,R-1} & \\
\hline
 & c_1 & c_2 & \cdots & c_{R-1} & c_R
\end{array}
$$

Table 3.2: Generic Butcher Tableau

For RK4, we then have the following Butcher Tableau:

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
1 & 0 & 0 & 1 & \\
\hline
 & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6}
\end{array}
$$

Table 3.3: RK4 Butcher Tableau

For more information regarding numerical differential equations solvers, refer to [6].

3.1.5.2 Dormand-Prince Method.

Given a problem in the form of equations (3.21-3.22), MATLAB's ODE45 uses the Runka-Kutta(4,5) method, also known as the Dormand-Prince method, to obtain a solution. The Dormand-Prince method produces a fifth order solution while using a fourth-order estimate of the error [25]. In this section, we discuss the Dormand-Prince method and how it uses an adaptive step size technique in order to achieve specified error tolerances. We wish to use this built in package since it will allow for easy replication and application for future problems based on the research in this thesis. Moreover, ODE45 is faster and more accurate than other built in solvers [25].

The Dormand-Prince Method can be summarized with the Butcher Tableau in Table 3.4. The first row of c coefficients give the fifth-order solution and the second row gives the fourth-order solution.

0								
$\frac{1}{5}$	$\frac{1}{5}$							
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$						
$\frac{4}{5}$	$\frac{44}{45}$	$\frac{-56}{15}$	$\frac{32}{9}$					
$\frac{8}{9}$	$\frac{19372}{6561}$	$\frac{-25360}{2187}$	$\frac{64448}{6561}$	$\frac{-212}{729}$				
1	$\frac{9017}{3168}$	$\frac{-355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$\frac{-5103}{18656}$			
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$\frac{-2187}{6784}$	$\frac{11}{84}$		
	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$\frac{-92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$	Fifth Order
	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$\frac{-2187}{6784}$	$\frac{11}{84}$	0	Fourth Order

Table 3.4: Dormand-Prince Butcher Tableau

Thus we see that the fifth-order solution can be obtained by iterating using the following equation:

$$\tilde{\boldsymbol{x}}_{j+1} = h(\frac{5179}{57600}\boldsymbol{K_1} + \frac{7571}{16695}\boldsymbol{K_3} + \frac{393}{640}\boldsymbol{K_4} - \frac{92097}{339200}\boldsymbol{K_5} + \frac{187}{2100}\boldsymbol{K_6} + \frac{1}{40}\boldsymbol{K_7})$$

and that the fourth-order solution can be obtained by iterating using the following equation:

$$\boldsymbol{x}_{j+1} = h(\frac{35}{384}\boldsymbol{K_1} + \frac{500}{1113}\boldsymbol{K_3} + \frac{125}{192}\boldsymbol{K_4} - \frac{2187}{6784}\boldsymbol{K_5} + \frac{11}{84}\boldsymbol{K_6})$$

We can then approximate the local truncation error as $||\tilde{\boldsymbol{x}}_{j+1} - \boldsymbol{x}_{j+1}||$. The procedure for implementing an adaptive step size technique is summarized in the steps below.

1. h given as initial step size, ϵ given as error tolerance

2. $t_{j+1} = t_j + h$

3. Calculate $\tilde{\boldsymbol{x}}_{j+1}$ and $\boldsymbol{x}_{j+1}$

4. If $||\tilde{\boldsymbol{x}}_{j+1} - \boldsymbol{x}_{j+1}|| \leq \epsilon h$, iterate using current step size h. In this case, we see that the approximated local error meets our error tolerance, hence we iterate without changing step size.

5. If criteria in step 4 is not met, let $q = \left(\frac{\epsilon h}{2||\tilde{\boldsymbol{x}}_{j+1} - \boldsymbol{x}_{j+1}||}\right)^{1/4}$. In this case, q is a scaling factor. We wish to determine qh that will meet our error tolerance. Specifically, qh must satisfy $\epsilon > \frac{||\tilde{\boldsymbol{x}}_{j+1} - \boldsymbol{x}_{j+1}||}{qh}$. That is, we need
   $\epsilon > \frac{||\tilde{\boldsymbol{x}}_{j+1} - \boldsymbol{x}_{j+1}||}{qh} = O((qh)^4) = kq^4h^4 = \frac{q^4}{h}||\tilde{\boldsymbol{x}}_{j+1} - \boldsymbol{x}_{j+1}||$ since $||\tilde{\boldsymbol{x}}_{j+1} - \boldsymbol{x}_{j+1}|| = O(h^5)$.
   Hence, we need $\epsilon > \frac{q^4}{h}||\tilde{\boldsymbol{x}}_{j+1} - \boldsymbol{x}_{j+1}||$, finally giving us, $q < \left(\frac{\epsilon h}{||\tilde{\boldsymbol{x}}_{j+1} - \boldsymbol{x}_{j+1}||}\right)^{1/4}$.
   Furthermore, we incorporate a "safety" factor of $\frac{1}{2}$ to get $q = \left(\frac{\epsilon h}{2||\tilde{\boldsymbol{x}}_{j+1} - \boldsymbol{x}_{j+1}||}\right)^{1/4}$

6.
   - If $q \leq 0.1$, let $h = .1h$
   - If $.1 < q \leq 4$, let $h = qh$
   - If $q > 4$, let $h = 4h$
   - If $h > h_{max}$, let $h = h_{max}$, where h_{max} is the maximum allowed step size

7. Iterate until desired time is achieved

The criteria in step 6 may be altered as desired. More information on adaptive time step methods can be found in [6] [17].

MATLAB's ODE45 can use the Dormand-Prince method and output solutions at given time steps. In our nonlinear model, we use integer values for our time discretization to represent the period of time between the data points we gather (days, weeks, months). Thus, we use MATLAB's ODE45 to provide solutions to our problem:

$$\begin{aligned} \boldsymbol{x}'(t) &= \boldsymbol{f}(t, \boldsymbol{x}) \\ \boldsymbol{x}(t_0) &= \boldsymbol{x}_0 \qquad a \leq t \leq b \end{aligned}$$

where $\boldsymbol{f}(t, \boldsymbol{x})$ is defined by equations (3.16 – 3.20), $\boldsymbol{x}_0$ are the initial values of $C(t)$, $E(t)$, $D(t)$ and $F(t)$ and $[a, b]$ is the time period which we solve our problem for.

3.1.6 Convergence of ODE45.

We now wish to determine why we believe the answers obtained using numerical differential equations solvers. To do this, we investigate the convergence of solutions provided by ODE45.

Definition 3.8. A method is convergent if the error, $||E_N|| \rightarrow 0$ as $h \rightarrow 0$ for any initial data y_0. In this case, $||E_N|| = ||\tilde{y}^N - y^N||$, where y^N is the solution obtained by the numerical differential equation solver at the final time $T = Nh$ and where $\tilde{y}^N$ is the exact solution at final time T.

Since y^n is the solution obtained by our solver, we see that by the definition of truncation error, τ_h, we have $\tilde{y}^{n+1} = H(\tilde{y}^n) + h\tau_h$, where H is our scheme (such as RK4 defined in Example 3.7). In the case of ODE45 where we have adaptive time steps, we have $\tilde{y}^{n+1} = H(\tilde{y}^n) + h_n\tau_{h,n}$, where h_n is the timestep taken at the n-th iteration and $\tau_{h,n}$ is the corresponding truncation error with the given timestep (since ODE45 solutions are fifth order accurate, $\tau_{h,n}$=$O(h_n^5)$). Thus we have:

$$\begin{aligned} ||E_N|| &= ||\tilde{y}^N - y^N|| \\ &= ||H(\tilde{y}^{N-1}) - H(y^{N-1}) + h_{N-1}\tau_{h,N-1}|| \\ &\leq ||H(\tilde{y}^{N-1}) - H(y^{N-1})|| + ||h_{N-1}\tau_{h,N-1}|| \end{aligned}$$

Applying the multivariate mean value theorem,

$$H(\tilde{y}^{N-1}) = H(y^{N-1}) + J(H)(\xi^{N-1})(\tilde{y}^{N-1} - y^{N-1})$$

where $J(H)(\xi^{N-1})$ is the Jacobian of H evaluated at ξ^{N-1} which lies between $\tilde{y}^{N-1}$ and y^{N-1}. Denote $\sigma_{N-1} = \|J(H)(\xi^{N-1})\|$. We then have :

$$\begin{aligned}
\|E_N\| &\leq \|J(H)(\xi^{N-1})(\tilde{y}^{N-1} - y^{N-1})\| + \|h_{N-1}\tau_{h,N-1}\| \\
&\leq \sigma_{N-1}\|(\tilde{y}^{N-1} - y^{N-1})\| + \|h_{N-1}\tau_{h,N-1}\| \\
&\leq \sigma_{N-1}\|E_{N-1}\| + \|h_{N-1}\tau_{h,N-1}\| \\
&\leq \sigma_{N-1}\sigma_{N-2}\|(\tilde{y}^{N-2} - y^{N-2})\| + \sigma_{N-1}\|h_{N-2}\tau_{h,N-2}\| + \|h_{N-1}\tau_{h,N-1}\| \\
&\vdots \\
&\leq \sigma_{N-1}\sigma_{N-2}...\sigma_0\|(\tilde{y}^0 - y^0)\| + \sigma_{N-1}\sigma_{N-2}...\sigma_1\|h_0\tau_{h,0}\| + ... + \sigma_{N-1}\|h_{N-2}\tau_{h,N-2}\| + \|h_{N-1}\tau_{h,N-1}\|
\end{aligned}$$

At our initial time, $\tilde{y}^0 = y^0$, thus we have:

$$\begin{aligned}
\|E_N\| &\leq \sigma_{N-1}\sigma_{N-2}...\sigma_1\|h_0\tau_{h,0}\| + ... + \sigma_{N-1}\|h_{N-2}\tau_{h,N-2}\| + \|h_{N-1}\tau_{h,N-1}\| \\
&\leq \sum_{j=1}^{N-1}\left(\left(\prod_{q=1}^{j}\sigma_{N-q}\right)\|h_{N-j-1}\tau_{h,N-j-1}\|\right) + \|h_{N-1}\tau_{h,N-1}\|
\end{aligned}$$

Here, we see that $\tau_{h,n}$ represents how consistent our calculations are at the n-th iteration while the σ_n terms, are the growth factors of the error at the n-th iteration. This represents the stability of our solution.

We now take a look at σ_n obtained from our system of differential equations and ODE45. Since, we have four equations in our system of differential equations, we will compute four sets of sigma values, $\sigma_n^{(m)}$, for $m = 1, 2, 3, 4$ and $n = 1...N - 1$. Using standard linear stability analysis (see [6]), we can obtain $\sigma_n^{(m)}$ if we have the eigenvalues of the Jacobian of $\boldsymbol{f}(\boldsymbol{x})$ evaluated at y^n, where $\boldsymbol{f}(\boldsymbol{x})$ is the right hand side of our system of differential equations and if we have our time steps taken at iteration n. For example, using RK4, $\sigma_n^{(m)}$ is computed with:

$$\sigma_n^{(m)} = 1 + \tfrac{h_n}{6}(\lambda_n^m + 2\lambda_n^m(1 + \tfrac{h_n}{2}\lambda_n^m) + 2\lambda_n^m(1 + \tfrac{h_n}{2}\lambda_n^m(1 + \tfrac{h_n}{2}\lambda_n^m)) + \lambda_n^m(1 + h_n\lambda_n^m(1 + \tfrac{h_n}{2}\lambda_n^m(1 + \tfrac{h_n}{2}\lambda_n^m))))$$

where λ_n^m is the m-th eigenvalue of the Jacobian of $\boldsymbol{f}(\boldsymbol{x})$ evaluated at y^n and h_n is the time step takin at the $n-th$ iteration. The $\sigma_n^{(m)}$ values for ODE45 are calculated similarly, using the Butcher tableau for the Dormand-Prince method [17], in place of the RK4 Butcher Tableau.

Running ODE45 to solve the problem given in equations (3.16-3.20) with a random set of parameters and initial values, we obtain the following $\sigma_n^{(m)}$ values in Figure 3.4. We do not observe $\sigma_m^{(3)}$ and $\sigma_m^{(4)}$. Given the nature of our system of differential equations, $\lambda_n^3 = \lambda_n^4 = 0$ for $n = 1, ..., N-1$. Hence $\sigma_n^{(3)} = \sigma_n^{(4)} = 1$ which is on the boundary of our stability region for Runge-Kutta schemes.

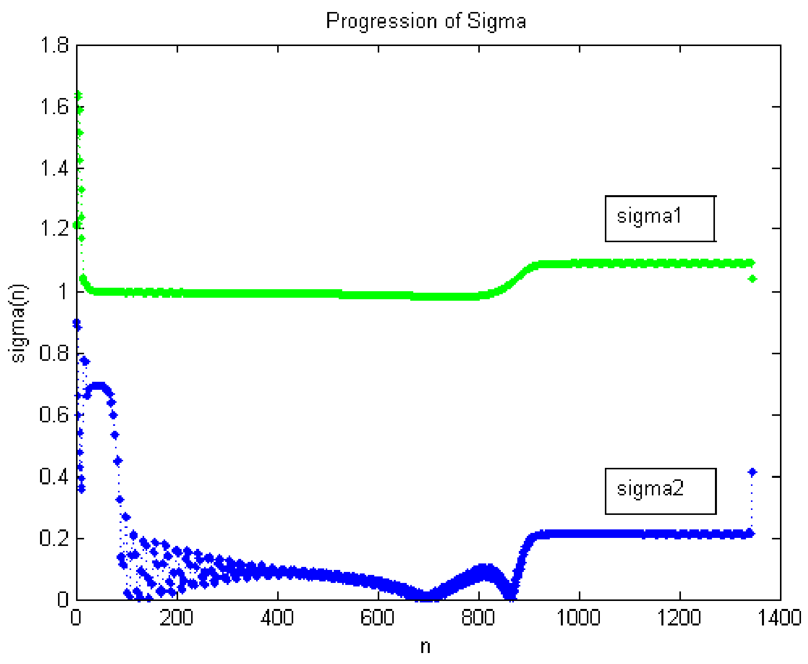


Figure 3.4: Sigma Values for ODE45 Run. We plot the two σ_n values for each time step taken. Here ODE45 outputs the solution for roughly $N = 1400$ time points. In this case, the first set of sigma values are greater than 1 or remain close to 1. For a convergent method, we wish to observe growth rates, σ_n, less than 1. We see that our $\sigma_n^{(1)}$ values may lead our inequality constraint to not meet convergence criteria.

Calculating the sum product, we have:

$||E_N|| \leq 1.2e10$

$||E_N|| \leq 5.3e-10$

for the solutions of the first and second equations in our system, respectively. Clearly, this calculation does not give us adequate error bounds for the solutions to our first equation. We must observe convergence indirectly. To resolve this matter, we perform the following steps:

1. Solve the problem implementing our own code (running RK4)
2. Calculate the sum product inequality for our implementation of RK4 and observe adequate inequality constraints for the RK4 solutions
3. Compare the solutions from our implementation of RK4 to the results of ODE45

Implementing an RK4 solver with a fixed time step h=5e-5, we obtain the following inequality constraint:

$$\begin{aligned}
||E_N|| &\leq ||J(H)(\xi^{N-1})(\tilde{y}^{N-1} - y^{N-1})|| + ||h\tau_h|| \\
&\leq \sigma_{N-1}||(\tilde{y}^{N-1} - y^{N-1})|| + ||h\tau_h|| \\
&\leq \sigma_{N-1}||E_{N-1}|| + ||h\tau_h|| \\
&\leq \sigma_{N-1}\sigma_{N-2}||(\tilde{y}^{N-2} - y^{N-2})|| + \sigma_{N-1}||h\tau_h|| + ||h\tau_h|| \\
&\vdots \\
&\leq \sigma_{N-1}\sigma_{N-2}...\sigma_0||(\tilde{y}^0 - y^0)|| + \sigma_{N-1}\sigma_{N-2}...\sigma_1||h\tau_h|| + ... + \sigma_{N-1}||h\tau_h|| + ||h\tau_h||
\end{aligned}$$

At our initial time, $\tilde{y}^0 = y^0$, thus we have:

$$\begin{aligned}
||E_N|| &\leq \sigma_{N-1}\sigma_{N-2}...\sigma_1||h\tau_h|| + ... + \sigma_{N-1}||h\tau_h|| + ||h\tau_h|| \\
&\leq \left(\left(\sum_{j=1}^{N-1}\prod_{q=1}^{j}\sigma_{N-q}\right) + 1\right)||h\tau_h||
\end{aligned}$$

Supposing our method is consistent with $\tau_h = O(h^p)$, then we have convergence if

$$\left(\left(\sum_{j=1}^{N-1} \prod_{q=1}^{j} \sigma_{N-q}\right) + 1\right) \leq o(h^{-(p+1)})$$

.

Calculating the sum product term for the RK4 implementation ($p = 4$), we obtain:
$\left(\left(\sum_{j=1}^{N-1} \prod_{q=1}^{j} \sigma_{N-q}^{(1)}\right) + 1\right) = 9.6e14 = O(\frac{1}{h^3}) \leq o(h^{-(p+1)})$

$\left(\left(\sum_{j=1}^{N-1} \prod_{q=1}^{j} \sigma_{N-q}^{(2)}\right) + 1\right) = 1.0e3 = O(N) = O(\frac{1}{h}) \leq o(h^{-(p+1)})$

Thus, our implementation of RK4 satisfies the convergence criteria. Moreover, in this case, we get the following inequality constraints:

$||E_N|| \leq 3.0e-7$

$||E_N|| \leq 3.2e-19$

Thus, our solutions at final time T are sufficiently close to the exact solutions.

We must now compare the results of our two solvers. To do this, we take the difference of solutions from the two solvers using decreasing time steps for the RK4 implementation. To observe convergence, our RK4 solution must converge to the ODE45 solution as we decrease the timestep used for our RK4 implementation. Figure 3.5 illustrates the change in the difference between our two solutions, denoted $E(h) = ||y_{ODE45}^N(h) - y_{RK4}^N(h)||$, as we decrease our timestep for RK4.

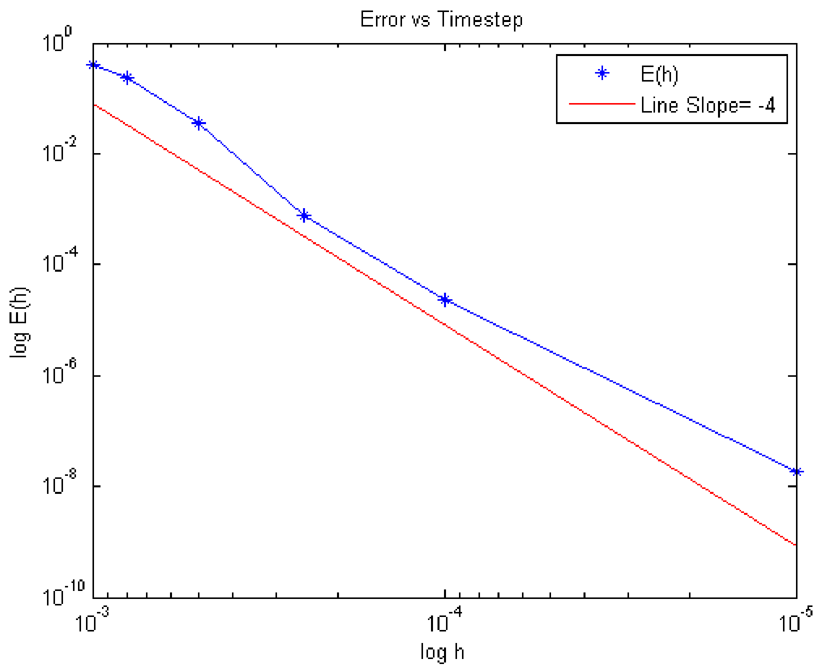


Figure 3.5: Difference Between RK4 and ODE45 Solutions as a Function of Time Step. Here, we plot the $log(E(h))$ over the $log(h)$ along with a line of slope= -4. We observe that as we decrease h for our implementation of RK4, the solution converges to the solution from ODE45. The *loglog* plot demonstrates that the RK4 solution converges at slope of -4 which is explained by the fact that RK4 is consistent on the order of $O(k^4)$.

Thus, we observe that our solutions from RK4 converges to the ODE45 solution at the rate of $O(k^4)$. Hence, the solutions to equations (316-3.20) using ODE45 are in fact convergent.

This section illustrated the criteria necessary to test convergence of our numerical schemes. We implemented our own RK4 scheme, observed convergence and then showed it converges to the ODE45 solution, thus showing that ODE45 is convergent. For more information on the convergence of numerical methods, refer to [6].

3.2 Data Fitting and Least Squares

Now that we have developed and solved the models that illustrates the behavior of the IED lifespan, we aim to employ the model to learn where IEDs are, at a given time and how fast they are moving. To do this, we will fit our model to data gathered from the battlefield.

From the battlefield, we can obtain sample points for both the number of IEDs that have detonated, D, and the number of IEDs that we find, F, at given times, however, we do not know how many IEDs are being constructed or emplaced. We also do not know the rates at which enemy forces are moving IEDs. Solving the least squares problem gives us an estimate on the entire supply chain. Suppose we cumulatively track the number of IEDs that detonate and the number of IEDs that are found at a given location for a certain period. Then, we see that $D(t)$ and $F(t)$ are monotonically increasing functions of time and our goal is to apply a least squares method on our data points to obtain the best approximation of our parameters. To do this, we use nonlinear least squares data fitting.

3.2.1 Nonlinear Least Squares Example.

A nonlinear least squares problem is a minimization problem of the form

$$\min_{x} \quad f(x) = \sqrt{\sum_{i=1}^{n} f_i(x)^2}$$

where the objective function $f(x)$ is defined in terms of f_i. This method is called least squares due to the fact that we minimize the sum of squares of these functions [12]. In our least squares problem, x will be represented by our parameter space.

One example of the application of the method of least squares can be seen when using data points to estimate parameters in mathematical biology. For example, suppose we collected data points (t_i, y_i) for $i = 1...n$ which represented the size of a population of antelope at various times. Suppose we have the data in Table 3.5.

t_i	1	2	4	5	8
y_i	3	4	6	11	20

Table 3.5: Antelope Population Data

where times are measured in years and population in hundreds. From population modeling, we approximate the population as an exponential model

$$y_i \approx x_1 e^{x_2 t_i}$$

where our parameters are x_1 and x_2. This model is illustrated in the figure below [12].

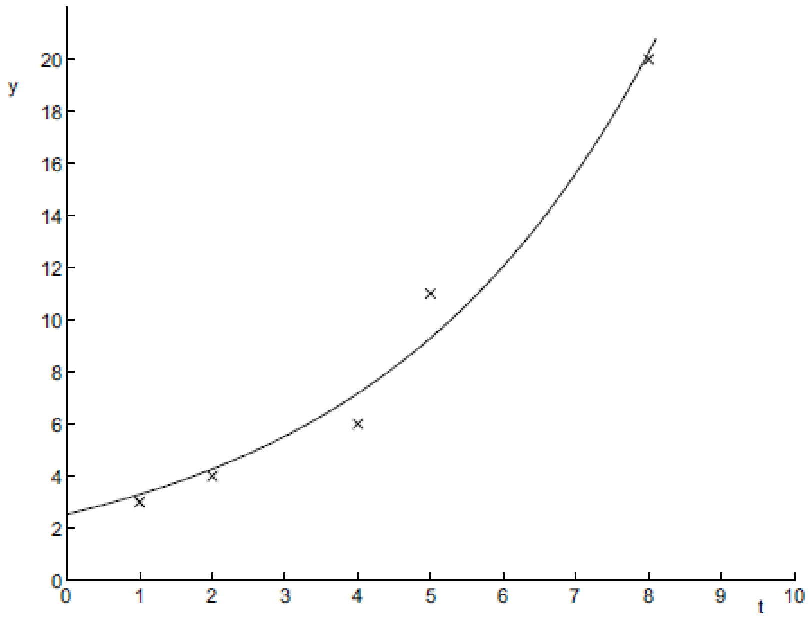


Figure 3.6: Antelope Population Model

Now using the least squares method, we set up the problem as

$$\underset{x}{\text{minimize}} \quad f(x_1, x_2) = \sqrt{\sum_{i=1}^{5}((x_1 e^{x_2 t_i}) - y_i)^2}$$

After solving for our optimal parameters x_1 and x_2 we report our error. In this case we see that at each data point we have

$$y_i = x_1 e^{x_2 t_i} + \epsilon_i$$

The error is then reported as

$$\text{Error} = \sqrt{\sum_{i=1}^{5} (\epsilon_i)^2}$$

For more information regarding nonlinear least squares, refer to [12].

3.2.2 Applying Nonlinear Least Squares.

Now, we can apply the least squares method to our IED model. If we sampled the number of IEDs destructed and found at $n + 1$ time points, we have the following data set:

- d_i
- $f_i \qquad for\; i = 0, 1, ...n$

where d_i and f_i represents the cumulative number of IEDs detonated and found, respectively at time i.

In our case, we must apply a least squares method to obtain the best parameters for the IED supply chain model. Applying the least squares method, we have the following objective function

$$\mathrm{m}(\overline{\sigma}) = \sqrt{\textstyle\sum_{i=0}^{n}[(D(t = i, \overline{\sigma}) - d_i)^2 + (F(t = i, \overline{\sigma}) - f_i)^2]}$$

where $\overline{\sigma}$ is a vector consisting of our parameters. In the linear case, $\overline{\sigma} =< k_1, k_2, k_3, k_4, \alpha_0, \alpha_1, \alpha_2, \beta_1, \beta_2 >$ and in the nonlinear case, we have $\overline{\sigma} =< C^*, \alpha_0, \alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2, \gamma_3, C(0), E(0), D(0), F(0) >$. Thus we have a nine-dimensional and a thirteen-dimensional least squares problem.

After solving for $\overline{\sigma}$ we then have all the data needed in order to solve for $C(t)$, $D(t)$, $E(t)$ and $F(t)$ for all time, thus obtaining an estimate for the entire supply chain behavior.

3.3 Local Optimization

In order to solve the least squares problem we defined in the section above, we must find a way to minimize our objective function, m($\overline{\sigma}$), over possible parameter values. To solve this minimization problem, we employ MATLAB's `lsqnonlin` tool [23]. To use `lsqnonlin`, users input the vector valued function: $\boldsymbol{p_i}(\boldsymbol{x})$=

$$\begin{pmatrix} p_1(\boldsymbol{x}) \\ p_2(\boldsymbol{x}) \\ \vdots \\ p_n(\boldsymbol{x}) \end{pmatrix}$$

Then, the problem is the restated as the following optimization problem:

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|p(\boldsymbol{x})\|_2^2 = \underset{\boldsymbol{x}}{\text{minimize}}(p_1(\boldsymbol{x})^2 + p_2(\boldsymbol{x})^2 + ... + p_n(\boldsymbol{x})^2)$$

The command `x=lsqnonlin(fun,x0)` starts at the intial guess point `x0` and finds a minimum of the sum of squares of the function described in `fun`. Note, '`fun`' is the vector valued function, $\boldsymbol{p}(\boldsymbol{x})$, and should return a vector of values, which MATLAB implicitly squares and sums up. Users can also input upper and lower bounds for the parameter space as well as other options. Refer to MATLAB's `lsqnonlin` documentation for more details [23].

To solve the nonlinear least squares problem, `lsqnonlin` uses a trust region reflective algorithm [27]. This algorithm only solves systems that are not underdetermined, that is there are at least as many data points as unknown parameters. Many methods of the MATLAB Optimization Toolbox are based on trust regions. To understand the trust region approach, let us consider minimizing $f(\boldsymbol{x})$. Supposing we are at a point $\boldsymbol{x}$, in n-space and wish to iterate and improve our function value. The goal is to approximate f with a simpler function q, which reasonably reflects the behavior of the function f in neighborhood N around the point $\boldsymbol{x}$. This neighborhood is the trust region.

This is reasonable as for general, smooth nonlinear functions, local approximate models fit the original function. A trial step $\boldsymbol{s}$ is computed by minimizing(approximately) over N. This creates the trust region subproblem:

$$\underset{\boldsymbol{s}}{\text{minimize}} \quad [q(\boldsymbol{s}), \boldsymbol{s} \in N]$$

The current point is updated to $\boldsymbol{x} + \boldsymbol{s}$ if $f(\boldsymbol{x} + \boldsymbol{s}) < f(\boldsymbol{x})$. Otherwise, the current point remains unchanged and the trust region N shrinks. In this case, we see that our trust region resulted in an inaccurate approximation. Thus, we shrink the region and the trial step computation is repeated.

The key questions in the trust region reflective algorithm are how to choose and compute the approximating function q and how to choose and modify the trust region N. In the trust region method, the quadratic approximation, q, is defined by the first two terms of the Taylor approximation of f at x and the neighborhood N is usually either spherical or ellipsoidal in shape. We see by Taylor expanding to two terms, we get:

$$f(\boldsymbol{x} + \boldsymbol{s}) = f(\boldsymbol{x}) + \boldsymbol{s}^T g + \frac{\boldsymbol{s}^T H \boldsymbol{s}}{2}$$

Where H is the Hessian matrix associated with f at x, and g is the gradient of f at x. Thus, to find $(\boldsymbol{x} + \boldsymbol{s})$ such that $f(\boldsymbol{x} + \boldsymbol{s}) < f(\boldsymbol{x})$, the trust region subproblem then becomes

$$\underset{\boldsymbol{s}}{\text{minimize}} \quad [\boldsymbol{s}^T g + \frac{\boldsymbol{s}^T H \boldsymbol{s}}{2}]$$

such that $||D\boldsymbol{s}||_2 \leq \delta$. Where D is a positive diagonal scaling matrix and δ is a positive scalar. This inequality constraint ensures that we minimize within our trust region.

Algorithms for solving the equation above exist; however, they may involve the computation of a full eigensystem which requires time proportional to several factorizations of H. Therefore, we use a separate approach to solving the trust region problem outlined in MATLAB's Unconstrained Nonlinear Optimization Algorithms [27].

To solve the trust region problem, we restrict the trust-region problem to a two dimensional subspace S. The work now lies in computing the subregion. We define S as the linear space spanned by the vectors $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$, where $\boldsymbol{s_1}$ is in the direction of the gradient and $\boldsymbol{s_2}$ is either a solution to

$$H\cdot \boldsymbol{s_2} = \text{-g}$$

or

$$\boldsymbol{s_2}^T \cdot H \cdot \boldsymbol{s_2} < 0$$

Thus, we now minimize the trust region subproblem along the plane spanned by $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$. The iteration performed is then:

1. Formulate the two dimensional trust region-subproblem
2. Solve the equation above to determine the trail step $\boldsymbol{s}$
3. If $f(\boldsymbol{x} + \boldsymbol{s}) < f(\boldsymbol{x})$, then $\boldsymbol{x} = \boldsymbol{x} + \boldsymbol{s}$. Else, shrink the trust region and reiterate.
4. Adjust δ

These steps are repeated until we obtain one of the following stopping criteria:

- MaxFunEvals: Maximum number of function evaluations allowed has been reached
- MaxIter: Maximum number of iterations allowed has been reached
- TolFun: Lower bound for change in function value is reached
- TolX: Lower bound for norm of change in $\boldsymbol{x}$ is reached

If `lsqnonlin` stops by acheiving TolFun or TolX, the run is considered to have converged. For more information on the trust region reflective algorithm refer to [27].

3.4 Global Optimization

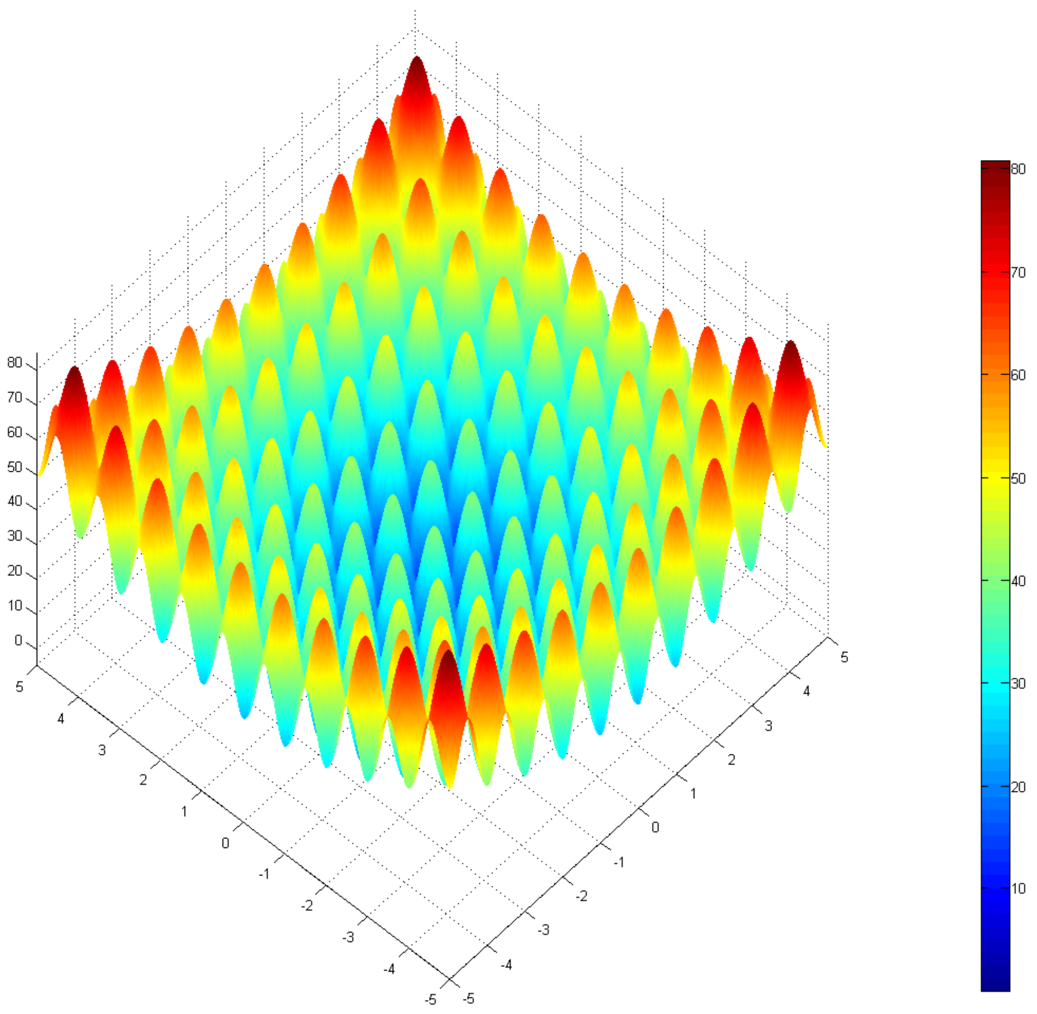
3.4.1 Introduction.

Using a local optimization solver gives us a look at what the minimizing parameter vector may be, however, the results may not be the most optimal fit to our data. It is possible that our `lsqnonlin` solver obtains a solution that represents the minimum of our objective function in only a local neighborhood. In order to arrive at the best parameter estimation, we need to implement a global optimization solver. This will ensure that we obtain the best fitting parameter estimate for our data.

3.4.2 Example(Rastrigin's Function).

To illustrate this behavior, consider the plot of Rastrigin's function in Figure 3.7 below [24]. The Rastrigin function is defined as:

$$Ras(x) = 20 + x_1^2 + x_2^2 - 10(\cos(2\pi x_1) + \cos(2\pi x_2))$$



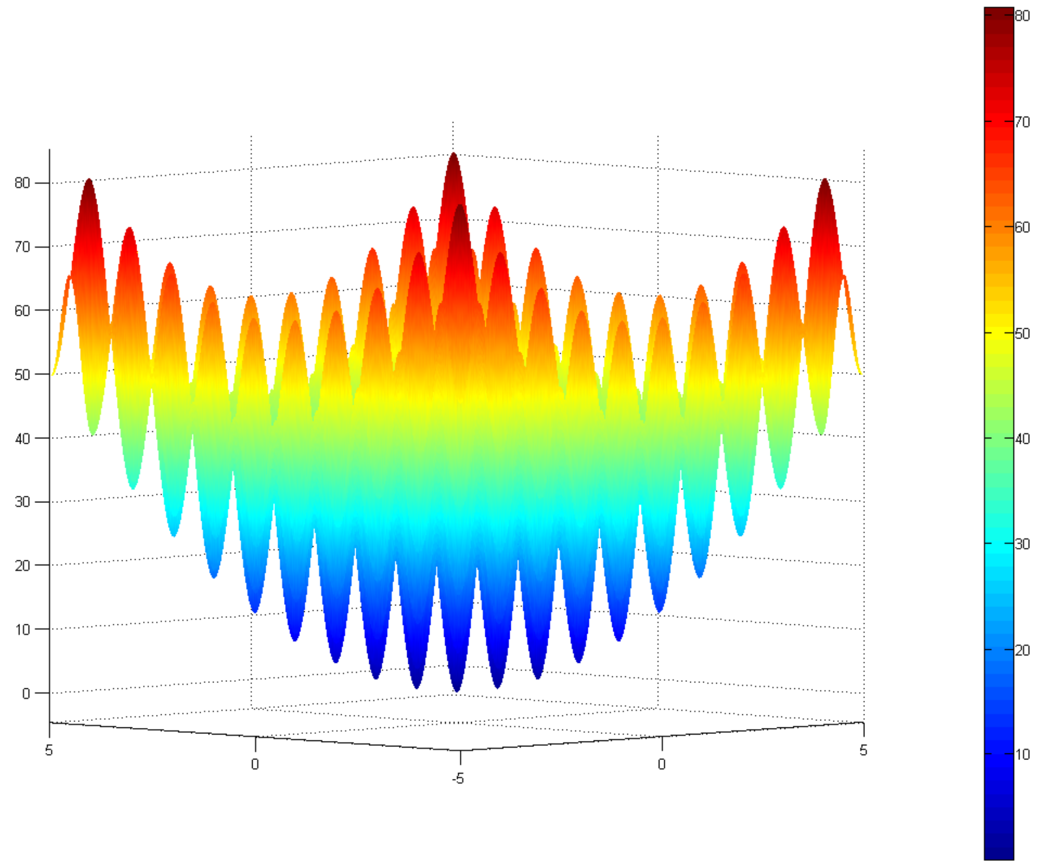


Figure 3.7: Rastrigin's Function

We see that Rastrigin's function contains many different local minima, represented by the valleys in the plot. However, this function has only one global minimum located at (0,0,0). At any of the other local minima, the function values are greater than 0. We also note that the farther away the local minima is from the origin, the larger the function value at the minima is.

Rastrigin's functions represents a good example of how a local optimization solver can fail to obtain correct results [24]. The existence of numerous different local minimas creates the possibility that our local optimization solver may give us a solution that is only the minimum in a given neighborhood and not the entire feasible region. For these reasons, we must implement a global optimization solver. For this thesis, we implement three different global optimization solvers.

3.4.3 MultiStart.

The first global optimization solver we use is MATLAB's MultiStart tool [20]. MultiStart takes a local solver, in our case the `lsqnonlin` solver, and intializes it from multiple different starting guesses. The aim of this global optimization solver is to sample different basins of attraction. For a given local minimum, the basin of attraction is defined to be the set of initial guesses that lead to the said minimum when optimizing using `lsqnonlin`. MATLAB's MultiStart performs the following steps: generate starting points, run local solver, check stopping conditions and create GlobalOptimSolution object.

1. Generate Starting Points- With MATLAB's MultiStart, users can define the bounds for the starting points. From here, MATLAB randomly distributes a set of starting points to use as initial guesses. MultiStart can be used in parallel by distributing the starting points to different processors where the local solver is then executed.

2. Run Local Solver (`lsqnonlin`)- In this step, MATLAB runs `lsqnonlin` at the given starting point. When the local solver finishes running, MultiStart stores the solutions and moves to the next step. After each iteration, MATLAB checks to see if the stopping conditions have been met.

3. Check Stopping Conditons- two stopping criteria for MultiStart.

    - MaxTime- The maximum time, defined by the user, allotted for MultiStart to run
    - All starting points used

4. Create GlobalOptimSolution object- When the stopping conditions are met, MultiStart creates the GlobalOptimSolution object which contains:

    - The solutions obtained by running `lsqnonlin`
    - The function values at the solutions
    - The points in the basin of attraction for each solution

More information on MultiStart can be found in MATLAB documentation [18], [20], [22], [26].

3.4.4 GlobalSearch.

The next global optimization solver we implement is MATLAB's GlobalSearch tool [20]. GlobalSearch differs from MultiStart in that it is used to find a single global minimum on a single processor where as MultiStart is used to find multiple local minima and can run in parallel. GlobalSearch only runs `fmincon`, a local solver that finds the minimum of a constrained function of several variables. Similar to `lsqnonlin`, `fmincon` uses a trust region reflective algorithm to find the global minimum. The GlobalSearch algorithm is summarized in the following steps:

1. Run local solver, `fmincon`, from initial starting point `x0`.
2. Generate a set of trial points.
3. Run local solver at the best starting point among a subset of trial points.
4. Loop through the remaining trial points, running `fmincon` if the point satisfies constraint conditions as well as other specified conditions.
5. Create GlobalOptimSolutions vector.

A more detailed version of the GlobalSearch algorithm is listed below.

1. Run `fmincon` from `x0`:

   To use GlobalSearch, users must first input an objective function and a starting point. GlobalSearch runs `fmincon` from the starting point. If this results in convergence, the starting point and end point are used for an initial estimate on the radius of a basin of attraction. GlobalSearch then records the function value for use in a score function. In this case, the score function is the sum of the function value at a certain point and a multiple of the sum of the constraint violations. Hence, a feasible point has a score equal to its function value.

2. Generate Trial Points, Obtain Stage 1 Start Point, Run:

   From here, GlobalSearch generates a set of trial points using a scatter search algorithm. GlobalSearch then performs function evaluations at a subset of the trial points, taking the point with the best score, called the Stage 1 Start Point, to run `fmincon` on, the subset of trial points are then removed from the original set of trial points.

3. Initialize Basins, Counters, Threshold:

   The localSolverThreshold is initially the smaller of the two objective function values of the solution to `fmincon` at `x0` and the Stage 1 Start Point. The initial estimate for the basins of attraction for the solution of these two points are spheres centered at each point with the radius being the distance from the initial points to the solution points. GlobalSearch keeps track of two counters.

   - Number of consecutive trial points that lie within a basin of attraction. One for each basin.
   - Number of consecutive trial points that have a score function greater than localSolverThreshold

4. Begin Main Loop:

Global Search examines the remaining trial points, the Stage 2 Trial Points, peforming the following steps. MaxTime is monitored, and stops the search if MaxTime has elapsed.

5. Examine Stage 2 Trial Point to See if `fmincon` Runs:
   GlobalSearch calls trial point `p`, and runs `fmincon` starting at `p` if the following conditions hold:

   - `p` is not in any existing basin
   - the score of `p` is less than localSolverThreshold
   - `p` satisfies bound and/or inequality constraints defined by user

6. When `fmincon` Runs:

   - Reset counters for basins and threshold to zero
   - Update the solution set. That is, if the `fmincon` run from `p` converges, GlobalSearch updates the GlobalOptimSolution object
   - Update basin radius and threshold. That is, if `fmincon` converges, we set radius to the maximum of the distance between p and its solution and the existing radius. We update the threshold to the score value of p
   - Report to iterative display

7. When `fmincon` Does Not Run:

   - Update counters. Increment the counter for every basin containing p. Reset the counter of every other basin to zero. Also, increment the threshold counter if the score of `p` is greater than or equal to localSolverThreshold. Otherwise reset the counter to zero.

- React to Large Counter Values. For each basin with counter equal to MaxWaitCycle, we shrink the basin radius and reset the counter to zero. If the threshold counter equals MaxWaitCycle, we increase the threshold.

8. Report to Iterative Display. Every 200th trial points is displayed.

9. GlobalSearch has the same stopping criteria as MultiStart:

   - MaxTime- The maximum time, defined by the user, allotted for MultiStart to run

   - All starting points used

After reaching one of the stopping criteria, GlobalSearch creates the GlobalOptimSolution object which reports the solution and function value at the solution. More information of GlobalSearch can be found in [20]

3.4.5 Genetic Algorithm.

The final global optimization solver we use is MATLAB's Genetic Algorithm [21]. In general, a genetic algorithm is a method for solving both constrained and unconstrained optimization problems based on a selection process that mimics biological evolution. The algorithm iteratively modifies a population of individual solutions. In our case this population is a set of vectors whose entries are the nine or thirteen parameters used for our IED model. At each iteration, a genetic algorithm randomly selects individuals in the current population, using them as parents to produce children for the next generation. Over generations, the population evolves into an optimal solution.

Specifically, MATLAB's Genetic Algorithm performs the following steps:

1. Begin by creating a random initial population

2. The algorithm then creates a sequence of new populations. At each of these steps, the algorithm uses points in the current population to create the next population. To create the new population, Genetic Algorithms does the following:

    - Scores each point in the current population using their function value
    - Scales the scores to easily arrange members by order
    - Selects members, now called parents, based on their function value scores
    - Chooses a number of parents with lower function scores and labels them as elite. These individuals are passed to the next generation since they have the lowest function values
    - Produces children from parents. This can be done by making random changes, called mutation or by combining the vector entries of a pair of parents, called crossover.
    - Replaces population with children to form the next generation

3. The algorithm stops when one of the stopping criteria is met. There are numerous different options for stopping conditions for Genetic Algorithm.

    - Maximum Number of Generations Reached
    - Maximum Time
    - Fitness Limit - The algorithm stops when it attains a point with fitness value lower than or equal to Fitness Limit
    - Stall Generations/Function Tolerance - The algorithm stops when the weighted average change in fitness function value divided by Stall generations is less than Function Tolerance. Stall generations is defined by the consecutive number of generations in which the optimal function value has not decreased

- Stall Time Limit - The algorithm stops if there is no improvement during a period of time equal to Stall Time Limit

At each vector coordinate, the Genetic Algorithm randomly selects one of the parents' corresponding coordinate value to pass on to the child for the crossover case. For the mutation case, the algorithm creates mutation by randomly changing the coordinate values of the parents. Elite parents are passed down without any change. More detailed information on the Genetic Algorithm can be found in MATLAB documentation [19] [21].

IV. Results and Analysis

4.1 Solver Verification

To validate that our optimization methods work, we first use them on a test set of data. Specifically, we fix a vector of our linear model parameters, $\overline{\sigma}_0$, and use equations (3.8-3.11) to obtain data points. We then start our local and global optimization solvers from a random point to see if we obtain the correct results. Doing this, we obtain the following results with our linear model:

k_1	k_2	k_3	k_4	α_0	α_1	α_2	β_1	β_2
400	2055	373	682	100	.4	.4	.6	.8

Table 4.1: $\overline{\sigma}_0$. We use this parameter vector to generate the test data points used for solver verification.

Solver	Function Value at Minimum
lsqnonlin	3.27e-4
MultiStart	9.04e-11
GlobalSearch	8.49e-2
Genetic Algorithm	7.32e-3

Table 4.2: Linear Verification Results. Running our four minimization solvers, we obtain these function values.

The optimization solvers for this case returns $\overline{\sigma}_0$. Similarly, with our nonlinear model, we obtain the following results:

Solver	Function Value at Minimum
lsqnonlin	1.36e-2
MultiStart	1.33e-2
GlobalSearch	1.41e-2
Genetic Algorithm	1.34e-2

Table 4.3: Nonlinear Verification Results. Running our four minimization solvers, we obtain these function values.

The solution (MultiStart) in this case is:

$C(0)$	$E(0)$	$D(0)$	$F(0)$	C^*	α_0	α_1	α_2	β_1	β_2	γ_1	γ_2	γ_3
500	500	100	100	2.6e6	100	.4	.4	.6	.5	7.9e-10	7.7e-7	1.8e-6

Table 4.4: Nonlinear Verification Solution

Solving for k_1, k_2, k_3, k_4 from $C(0)$, $E(0)$, $D(0)$, $F(0)$ using equations (3.12-3.15), we also see that the solvers return the parameter values in $\overline{\sigma}_0$. Thus, we see that our solvers converge to $\overline{\sigma}_0$.

4.2 IED Data from Iraq

We can now apply all the methods previously mentioned in order to analyze IED supply chain behavior. An IED metrics report written by the Center for Strategic International Studies [2] contained the data displayed in Table 4.5 regarding the number of

IEDs that were detonated and found in Iraq over a 26 month period from April 2008 to May 2010. The information in Table 4.5 is also displayed in a plot in Figure 4.1. We use the following information as data for our least squares problem from here on.

Time (months)	Detonated (d_i)	Found (f_i)	Time(months)	Detonated (d_i)	Found (f_i)
0	93	710	13	434	4634
1	144	1244	14	460	4783
2	191	1615	15	469	4870
3	227	1958	16	480	4992
4	252	2293	17	491	5108
5	274	2617	18	503	5218
6	296	2905	19	510	5306
7	325	3218	20	519	5375
8	346	3485	21	529	5464
9	368	3744	22	541	5532
10	386	3977	23	554	5616
11	393	4206	24	565	5689
12	413	4437	25	580	5749

Table 4.5: IEDs Detonated and Found in Iraq

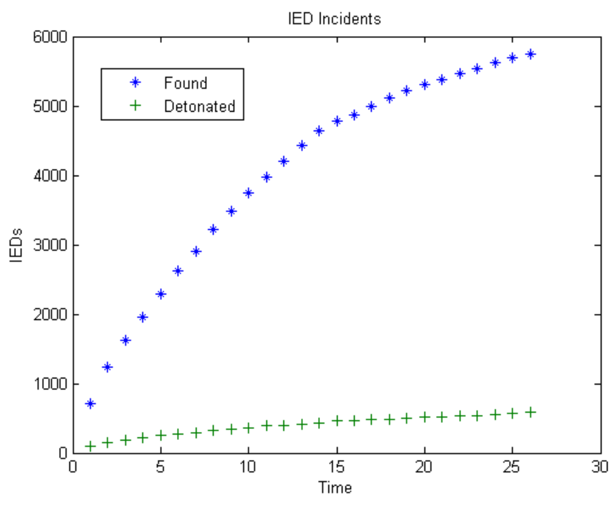


Figure 4.1: Plot of IEDs Detonated and Found

4.3 Linear Model Results

Formulating the least squares problem for our linear case, we obtain the following objective function:

$$m(\overline{\sigma})= \sqrt{\sum_{i=0}^{25}[(D(t=i,\overline{\sigma})-d_i)^2+(F(t=i,\overline{\sigma})-f_i)^2]} \tag{4.1}$$

where $\overline{\sigma}=< k_1, k_2, k_3, k_4, \alpha_0, \alpha_1, \alpha_2, \beta_1, \beta_2 >$ is a vector consisting of our nine parameters and where

$$D(t) = \left(\frac{\alpha_2\alpha_0\alpha_1}{r_1 r_2}\right)t + \frac{\alpha_2\alpha_1 k_1 e^{-r_1 t}}{r_1\left(r_1-r_2\right)} - \frac{\alpha_2 k_2 e^{-r_2 t}}{r_2} + k_3$$

$$F(t) = \left[\frac{\beta_1\alpha_0}{r_1} + \left(\frac{\beta_2\alpha_0\alpha_1}{r_1 r_2}\right)\right]t - \frac{\beta_1 k_1 e^{-r_1 t}}{r_1} + \frac{\beta_2\alpha_1 k_1 e^{-r_1 t}}{r_1\left(r_1-r_2\right)} - \frac{\beta_2 k_2 e^{-r_2 t}}{r_2} + k_4$$

Introducing the data listed in Table 4.5 for d_i and f_i for $i = 0, 1, 2, ..., 25$ into equation 4.1, we have our complete objective function.

4.3.1 Local Optimization Solutions.

To implement our local solver, we must first determine the parameter values to use as an initial guess. Suppose we make the initial guess of the rate parameters and initial values of each state listed in Table 4.6. Then, from equations (3.12-3.15), we obtain a parameter vector, $\overline{\sigma}$, as our initial guess to run `lsqnonlin`.

Parameter	$C(0)$	$E(0)$	$D(0)$	$F(0)$	α_0	α_1	α_2	β_1	β_2
Guess Value	2450	1680	93	710	150	.8	.1	.2	.65

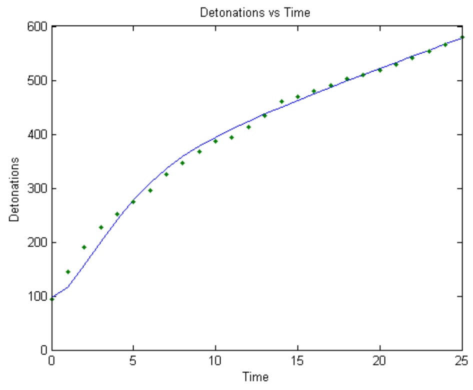
Table 4.6: Initial Guess Values for Linear Model

Using the local solver `lsqnonlin` we obtain the following function value as our minimum: $m(\overline{\sigma})$=932.0660. Our initial guess parameters, bounds and solutions are listed in the table below:

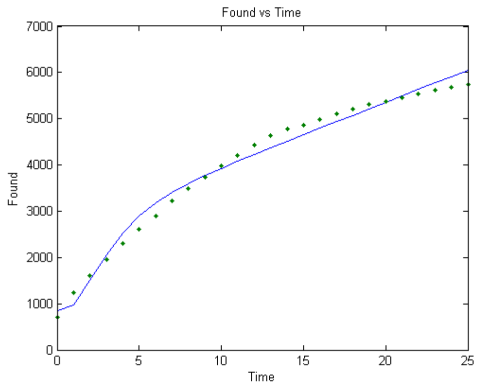
Parameter	Lower Bound	Upper Bound	Initial Guess	Solution
k_1	$-\infty$	∞	2300	3019.7
k_2	$-\infty$	∞	9360	1902421.4
k_3	$-\infty$	∞	-467	257.0
k_4	$-\infty$	∞	-3420	2577.3
α_0	0	∞	150	152.3
α_1	0	.99	.8	.579
α_2	0	.99	.1	.063
β_1	0	.99	.2	.063
β_2	0	.99	.65	.578

Table 4.7: Parameter Values for Linear `lsqnonlin` Solution

Figure 4.2 gives us the fit to our data obtained from our solution. We aim to obtain better solutions through our global optimization methods.



(a) IEDs Detonated Fit



(b) IEDs Found Fit

Figure 4.2: Data Fits for Linear Model Using Local Solver. This data fit obtains a function value of $m(\overline{\sigma})$=932.0660. We aim to improve our results using global optimization tools.

4.3.2 Global Optimization Solutions.

4.3.2.1 MultiStart.

We now apply our global optimization methods in order to find a better solution. Running the MultiStart global optimization solver, we were able to run `lsqnonlin` from 4,505 different starting points for our least squares problem. With this optimization tool, we obtain a minimum of $m(\overline{\sigma})$=175.2102. The table below shows the bounds of our random starting points as well as the parameter values of our solution.

Parameter	Lower Bound	Upper Bound	Initial Guess	Solution
k_1	$-\infty$	∞	2300	3544.5
k_2	$-\infty$	∞	9360	96499.8
k_3	$-\infty$	∞	-467	596.6
k_4	$-\infty$	∞	-3420	6127.8
α_0	0	∞	150	2.8e-11
α_1	0	.99	.8	.110
α_2	0	.99	.1	.013
β_1	0	.99	.2	.034
β_2	0	.99	.65	.127

Table 4.8: Parameter Values for Linear MultiStart Solution

4.3.2.2 GlobalSearch.

Running GlobalSearch, we obtain a minimum of $m(\overline{\sigma})$=177.1013.

Parameter	Lower Bound	Upper Bound	Initial Guess	Solution
k_1	$-\infty$	∞	2300	4210.6
k_2	$-\infty$	∞	9360	-2191.5
k_3	$-\infty$	∞	-467	466.4
k_4	$-\infty$	∞	-3420	4608.8
α_0	0	∞	150	55.37
α_1	0	.99	.8	.084
α_2	0	.99	.1	.048
β_1	0	.99	.2	.074
β_2	0	.99	.65	.275

Table 4.9: Parameter Values for Linear GlobalSearch Solution

4.3.2.3 Genetic Algorithm.

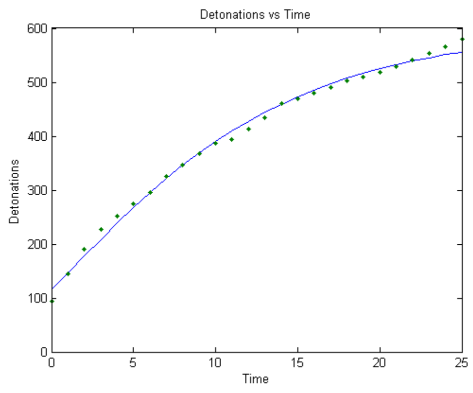
Finally, running Genetic Algorithm, we obtain a minimum of m($\overline{\sigma}$)=237.7092. However, since GeneticAlgorithm is typically used to obtain a general idea of where the minimum may exist, we run our local solver, `lsqnonlin`, starting at the Genetic Algorithm output to obtain a better solution. Thus, performing `lsqnonlin`, we obtain a minimum of m($\overline{\sigma}$)=175.2597 with the following values:

Parameter	Lower Bound	Upper Bound	Initial Guess	Solution
k_1	$-\infty$	∞	2300	3795.9
k_2	$-\infty$	∞	9360	-39647.7
k_3	$-\infty$	∞	-467	598.1
k_4	$-\infty$	∞	-3420	6129.0
α_0	0	∞	150	3.84e-14
α_1	0	.99	.8	.100
α_2	0	.99	.1	.015
β_1	0	.99	.2	.037
β_2	0	.99	.65	.132

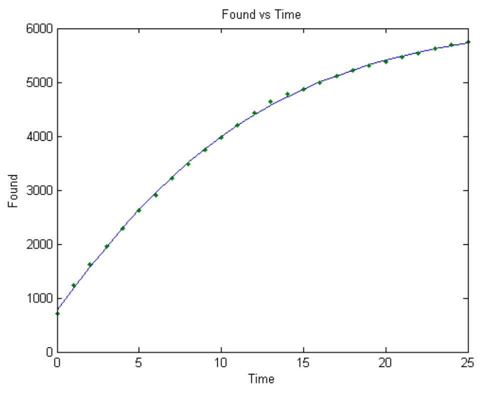
Table 4.10: Parameter Values for Linear Genetic Algorithm Solution

4.3.3 Final Results.

We see that the MultiStart optimization tool gives us the best fit parameters for our data. MultiStart obtains a function minimum of $m(\overline{\sigma})$=175.2102. The least squares function value gives us the absolute error, the difference between our solution and the exact data, however, if we wish to incorporate the magnitude of the exact solution, we should report the relative error. We define the relative error as $E = \frac{m(\overline{\sigma})}{||p||_2}$, where p is the vector containing our 52 data points. That is, the relative error is the absolute error divided by the magnitude of the exact solution. Thus, we have $E = 0.008$. The following plots illustrate the progression of the four states through the 26 month period. Here, we see that applying our least squares method, we can estimate how many IEDs were being constructed and emplaced at a give time in the past.

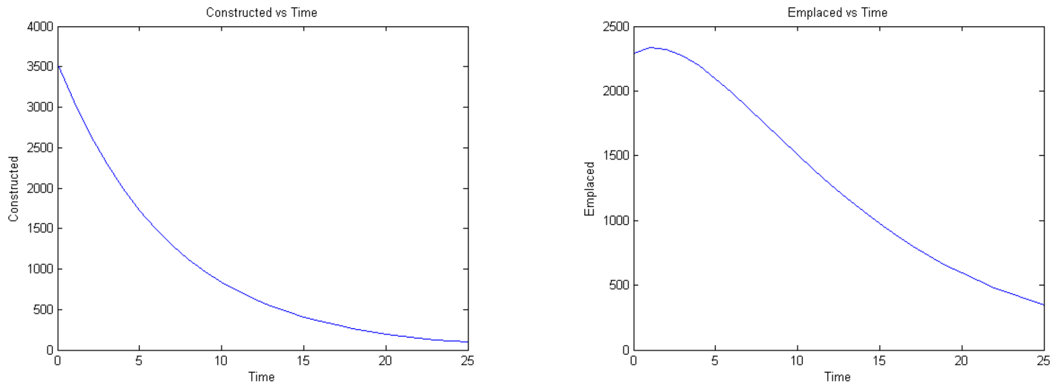


(a) IEDs Detonated Fit



(b) IEDs Found Fit

Figure 4.3: Data Fits for Linear Model Using Global Solver. This plot illustrates the best fit we obtain for our linear model. We achieve a relative error of $E = 0.008$. Furthermore, we see that the data points for IEDs detonated seem to oscillate around the obtained fit. This could be explained by a some periodic, time dependent occurence which may be accounted for in the nonlinear model.



(a) IEDs Constructed (b) IEDs Emplaced

Figure 4.4: Approximated Curves for Linear Model (MultiStart). To begin, we only had access to data on IEDs found and detonated. After solving the least squares problem, we now have an estimate for the numbers of IEDs being constructed and emplaced. In this case, we gain valuable information on the behavior of the IED supply chain. We observe that enemy forces are constructing fewer IEDs as time progresses. Furthermore, at $t \approx 2$, we observe that emplaced IEDs achieved a maximum. This means that at $t \approx 2$, coalition forces' efforts in defeating the device began to overpower enemy forces' efforts at emplacing IEDs.

Without this estimation, coalition forces would have to use the number of IEDs detonated and found as a metric to determine how well they are performing in attacking the network and defeating the device. After solving our least squares problem, we have a more direct metric for coalition forces to use. Since the number of IEDs in the construction state is in a constant state of decrease, we can conclude that coalition forces were adequately attacking the network for the duration of the 26 months. As for IEDs emplaced, we can conclude that at time $t \approx 2$, coalition forces were putting enough effort into defeating the device to begin winning against enemy forces' efforts at emplacing

IEDs. Thus, we see that modeling the supply chain and solving our least squares parameters offers a powerful tool at gauging coalition forces' C-IED efforts.

4.4 Nonlinear Model Results

We can now apply all the methods previously mentioned in order to analyze IED supply chain behavior for the nonlinear case. The least squares objective function for the nonlinear case is the same as the one used for the linear case:

$$\mathrm{m}(\overline{\sigma})= \sqrt{\sum_{i=0}^{25}[(D(t=i,\overline{\sigma})-d_i)^2+(F(t=i,\overline{\sigma})-f_i)^2]}$$

where in this case, $\overline{\sigma}=< C(0), E(0), D(0), F(0), C^*, \alpha_0, \alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2, \gamma_3 >$ is a vector consisting of our thirteen parameters and where $C(t)$, $E(t)$, $D(t)$, and $F(t)$ are given by the solutions obtained by running ODE45 on equations (3.16-3.20). We now perform the parameter estimation methods on the nonlinear case.

4.4.1 Local Optimization Solutions.

We aim to use the nonlinear model to improve our results from the linear model. To do, this we run `lsqnonlin` from an initial guess set to the minimum from the linear model. Recall from our linear model that the parameter values that gives us our best obtained minimum were:

Parameter	k_1	k_2	k_3	k_4	α_0	α_1	α_2	β_1	β_2
Value	3544.5	96499.8	596.6	6127.8	2.8e-11	.110	.013	.034	.127

Table 4.11: Linear Solution

Since we noted that our nonlinear model contains our linear model, we can take the optimized parameter values from the linear model and use them as the initial guess point for `lsqnonlin`. From equations (3.12-3.15), we see that we had the following initial values for $C(0)$, $E(0)$, $D(0)$, $F(0)$.

$C(0)$	$E(0)$	$D(0)$	$F(0)$
3544.5	2293.6	115.0	771.3

Table 4.12: Initial Function Values for Nonlinear Model

Hence, converting our optimized parameter values from the linear case into parameters for the nonlinear case we have the initial guess displayed in Table 4.13. We assign a large value to C^*.

$C(0)$	$E(0)$	$D(0)$	$F(0)$	C^*	α_0	α_1	α_2	β_1	β_2	γ_1	γ_2	γ_3
3544.5	2293.6	115.0	771.3	500000	2.8e-11	.110	.013	.034	.127	0	0	0

Table 4.13: Initial Guess for Nonlinear Model

Using this initial guess, `lsqnonlin` stops after 7 iterations due to the fact that the norm of the step size is less than TolX=1e-8. Our function value remains unchanged, $m(\overline{\sigma})$=175.2102. In this case, the nonlinearity imposed by γ_1, γ_2, γ_3, did not result in improvement of our function value. We now investigate the nonlinear model using global optimization methods.

4.4.2 Global Optimization Solutions.

4.4.2.1 MultiStart.

Running MultiStart, we obtain a minimum of $m(\overline{\sigma})$=174.8342 with the following values:

Parameter	Lower Bound	Upper Bound	Initial Guess	Solution
$C(0)$	0	10000	3544.5	4823.7
$E(0)$	0	10000	2293.6	1159.3
$D(0)$	0	300	115.0	129.5
$F(0)$	0	1000	771.3	774.7
C^*	1	5000000	500000	18365
α_0	0	1000	2.8e-11	.002
α_1	0	.99	.110	3.265e-10
α_2	0	.99	.013	.020
β_1	0	.99	.034	.054
β_2	0	.99	.127	.131
γ_1	0	.1	0	2.357e-10
γ_2	0	.1	0	4.399e-10
γ_3	0	.1	0	3.482e-4

Table 4.14: Parameter Values for Nonlinear MultiStart Solution

4.4.2.2 GlobalSearch.

Running GlobalSearch, we obtain a minimum of $m(\overline{\sigma})$=175.2096 with the following values:

Parameter	Lower Bound	Upper Bound	Initial Guess	Solution
$C(0)$	0	10000	3544.5	3591.0
$E(0)$	0	10000	2293.6	2246.8
$D(0)$	0	300	115.0	115.1
$F(0)$	0	1000	771.3	771.3
C^*	1	5000000	500000	499637
α_0	0	1000	2.8e-11	0
α_1	0	.99	.110	.108
α_2	0	.99	.013	.014
β_1	0	.99	.034	.034
β_2	0	.99	.127	.128
γ_1	0	.1	0	0
γ_2	0	.1	0	0
γ_3	0	.1	0	0

Table 4.15: Parameter Values for Nonlinear GlobalSearch Solution

4.4.2.3 Genetic Algorithm.

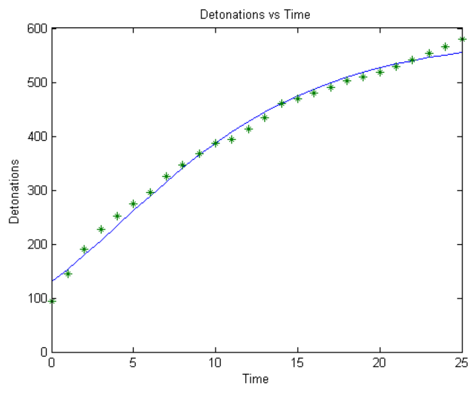
Running Genetic Algorithm and running `lsqnonlin`, we obtain $m(\overline{\sigma})$=175.2096 with the following values:

Parameter	Lower Bound	Upper Bound	Initial Guess	Solution
$C(0)$	0	10000	3544.5	3591.8
$E(0)$	0	10000	2293.6	2246.1
$D(0)$	0	300	115.0	115.1
$F(0)$	0	1000	771.3	771.3
C^*	1	5000000	500000	500000
α_0	0	1000	2.8e-11	3.92e-14
α_1	0	.99	.110	.108
α_2	0	.99	.013	.014
β_1	0	.99	.034	.034
β_2	0	.99	.127	.128
γ_1	0	.1	0	2.92e-14
γ_2	0	.1	0	2.95e-14
γ_3	0	.1	0	8.65e-10

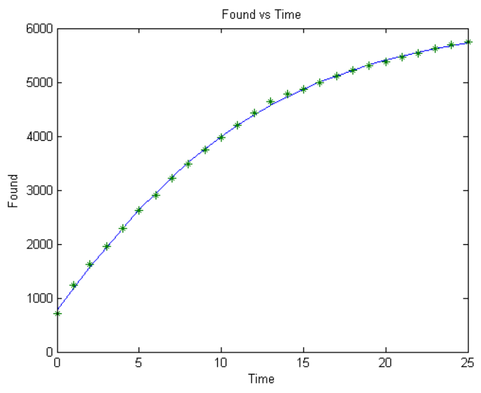
Table 4.16: Parameter Values for Nonlinear Genetic Algorithm Solution

4.4.3 Final Results.

We see that MultiStart once again gives us the best fit. MultiStart obtains a function minimum of $m(\overline{\sigma})$=174.8342. In terms of relative error, E=0.0079. The following plots illustrate the four states through the 26 month period.

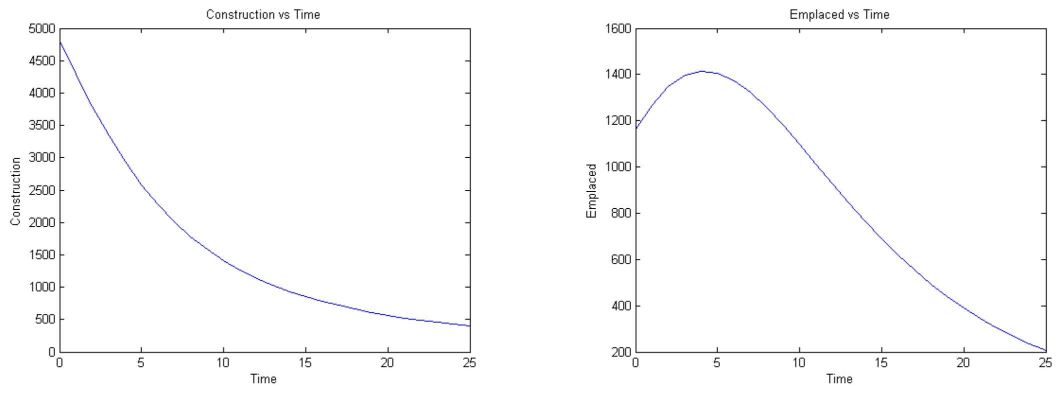


(a) IEDs Detonated Fit



(b) IEDs Found Fit

Figure 4.5: Data Fits for Nonlinear Model. This plot illustrates the best fit we obtain for our nonlinear model. We achieve a relative error of $E = 0.0079$. We again see the same oscillatory behavior for the IEDs detonated fit, thus, we can conclude that this behavior is independent the strategy changes we allow for in our model. This behavior must be explained by another source of time dependent oscillatory behavior, such as seasonal weather effects.



(a) IEDs Constructed (b) IEDs Emplaced

Figure 4.6: Approximated Curves for Nonlinear Model. With the new nonlinear model estimates, the approximated curves demonstrate the same behavior, however, we now have different vales for $C(0)$ and $E(0)$. We also see that the maximum for $E(t)$ occurs at $t \approx 5$.

4.5 Analysis

We used the nonlinear model to see if we can improve the minimum of our function from the linear model. By applying both local and global optimization methods, we saw that the improvements were marginal. We observed that although the nonlinear model did not significantly improve our function value from the least squares problem, the approximations for $C(t)$ and $E(t)$ that resulted were drastically different as pictured below.

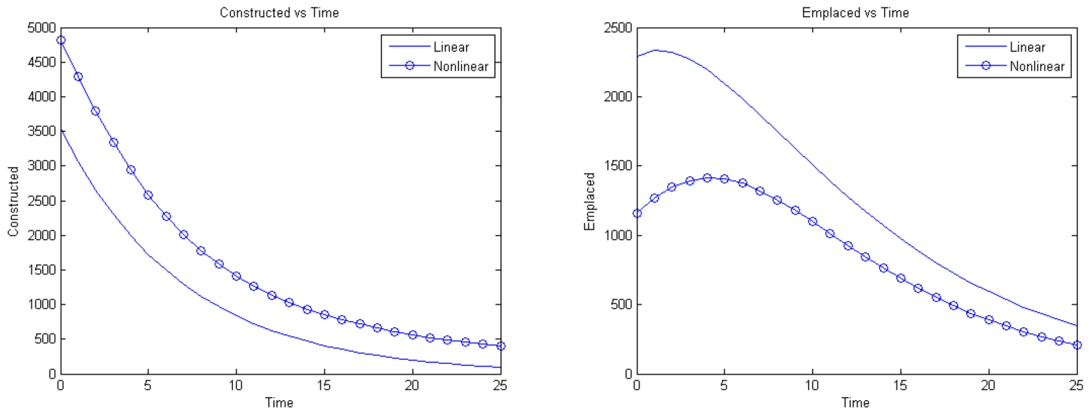


Figure 4.7: Estimate Comparison. Here we compare the two estimates for C and E that we obtain. The initial values, $C(0)$ and $E(0)$, changed drastically. The maximum of $E(t)$ also occured later in time for the nonlinear model, however, the general behavior of the two states were still similar. Both models gave us approximations where $C(t)$ was decreasing, while $E(t)$ achieved some maximum and then began decreasing, presumably when coalition forces' C-IED operations began to outweigh enemy forces' actions.

In this case, the two solutions present two unique minima for the objective function. As a result, these two minima provide different approximations for $C(t)$ and $E(t)$. Thus, although we obtained close fits to the observed data, the estimates for the unobserved behavior changed drastically. From this point, we must choose which solution to use for approximating the IED supply chain.

Looking at the best solution (MultiStart) from the nonlinear case, we see that the terms that impose the nonlinearity, γ_i, were close to zero, with the exception of $\gamma_3 = 3.482e - 4$. Moreover, we see that the critical capacity term we introduce, C^* does not impose nonlinearity; it is a linear term. Since the γ_i terms are close to zero, we may be inclined to conclude that the linear model may be sufficient in modeling the data we used. To investigate this, we now perform the least squares problem with the same data on the linear model with the same critical capacity term. This gives us the following linear system:

$$\begin{aligned}
\dot{C} &= \left(1 - \frac{C}{C^*}\right)\alpha_0 - (\alpha_1 + \beta_1)C \\
\dot{E} &= \alpha_1 C - (\alpha_2 + \beta_2)E \\
\dot{D} &= \alpha_2 E \\
\dot{F} &= \beta_1 C + \beta_2 E
\end{aligned}$$

Solving the least squares problem, we obtain $m(\overline{\sigma})$=175.2096. Recall that the best solution from the linear case was $m(\overline{\sigma})$=175.2102 and the best solution to the nonlinear case was $m(\overline{\sigma})$=174.8342. Thus, since we applied a least squares problem to the linear case including a capacity turn and still did not achieve as low of a minimum as the nonlinear case, we see that the nonlinearity imposed by γ_3 in our nonlinear solution was the cause for the more optimal minimum. Since the nonlinear terms did in fact provide better results and since the nonlinear model includes the linear model, we see that the nonlinear case here, does ultimately provide a better estimate.

We do note, however, that the linear model is more efficient in two ways. First, the linear model requires minimization over nine dimensions, while the nonlinear model requires minimization over thirteen dimensions. This causes our optimization methods to take more time. Second, in the nonlinear case we must run our numerical differential equations solver to obtain function values with given parameters at a given time. Since we have explicit solutions from equations (3.8-3.11) for the linear model, MATLAB only requires function evaluations to obtain function values. Thus, if efficiency were valued, we should use the linear model, with a critical capacity term.

We saw that we created a second, more complicated model, the nonlinear model to obtain better solutions to our data fits. The nonlinear model obtained marginally better fits for the observed data, but drastically different estimates for the unobservables. We saw the nonlinearity terms caused the change in the estimates. We choose to use the nonlinear estimates due to the fact that it did provide better data fits and due to the fact that it contains the linear model (thus allowing for further improvement of our function value). We also recognize though, that if our data does not behave similarly to the set used in this chapter, that is, if the number of IEDs detonated and found each time period are fluctuating, the nonlinear model may provide significantly more optimal solutions than the linear model as this would model enemy and coalition forces changing strategies.

One advantage of the nonlinear model is that it allows us to vary the rates over time. Hence, after solving the least squares problem for our nonlinear model, we can see how the rates in which IEDs flow from one state to other vary over time. We obtain the following rate functions using the parameter values that gave us the best function value:

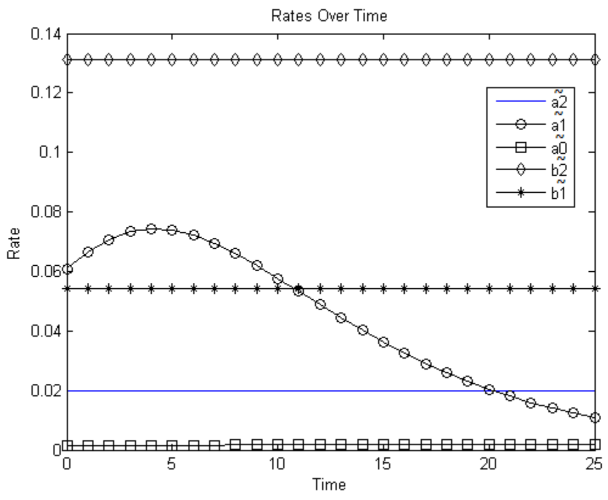


Figure 4.8: Rates over Time. Here, we observe the behavior of our flow rates over time. We see that the parameters remain nearly constant except for $\tilde{\alpha}_1$. This is expected as γ_1 and γ_2 were small compared to γ_3, which $\tilde{\alpha}_1$ was influenced by. The initial increase in $\tilde{\alpha}_1$ accounts for the difference between the linear and nonlinear model in the approximations for $E(t)$. The initial increase in $\tilde{\alpha}_1$ causes the maximum for $E(t)$ to occur later in time. Here, we see that the fluctuation in $\tilde{\alpha}_1$ is what allows us to obtain a better fit as the other rates remain constant.

V. Conclusions

5.1 Summary

To help combat IEDs, this thesis analyzed the IED supply chain process to provide new perspective on fighting IEDs. To accomplish this, we began by first developing a model of the supply chain terrorists use to develop, emplace and detonate IEDs. Our model contained four different states in which IEDs can exist in and five rate parameters which represented the flow of IEDs into and out of these states. For the flow rates, we considered two cases. The first case assumed that the flow rates were constant while the second case formulated flow rates which changed to represent basic strategies for both enemy and coalition forces.

In the case of constant flow rates we can obtain solutions explicitly. In the second case, we do not solve explicitly. Instead, we employed the use of a numerical differential equations solver. Along the way, we proved that solutions to our system of differential equations are unique. Furthermore, we created a way to show that MATLAB's ODE45 is convergent. To do this, we implemented our own RK4 scheme, observed convergence and then showed that our RK4 results converged to the ODE45 solutions.

From here, we applied a least squares method. Over a given period of time, coalition forces collect data on the number of IEDs that detonate as well as the number of IEDs they find. Using this data, we formulated a least squares problem. In the case of constant flow rates, we have a nine dimensional minimization problem while in the nonlinear case we have a thirteen dimensional minimization problem.

To solve our least squares problem, we then employed the use of optimization tools. We applied both local solvers (`lsqnonlin`) and global solvers (MultiStart GlobalSearch Genetic Algorithm). This then gave us optimal parameter sets for both cases. We observed that our nonlinear case contained our linear case, hence we applied the nonlinear case to improve our results from the linear case. Consequently, we observed that the improvements were marginal but the resulting estimates were quite different. We chose to use the nonlinear estimates as the nonlinear terms ultimately provided a better fit for the data. We do note that if the rate at which $D(t)$ and $F(t)$ increases fluctuates, the nonlinear model may provide substantial improvements as it allows for our rates to fluctuate.

Once the solution is obtained, we gain access to valuable information of the IED supply chain. To begin, coalition forces could only observe the number of IEDs detonated and found. After solving our least squares problem, not only do we know how many IEDs were detonated or found but we also can estimate the number of IEDs constructed and emplaced, as well as how fast they are being transported throughout the supply chain, this gives us an estimate on the behavior of the entire supply chain. In the data used from Iraq over a 26 month period, we estimated that IED construction was in decline the whole time. We also saw that our estimation showed an increase in emplaced IEDs followed by a decrease for the remainder of the time period. This allowed us to see when coalition efforts at defeating the device finally became adequate enough in order to begin winning the war against IEDs.

5.2 Future Work

In our model we did not consider a storage state for IEDs nor did we consider the fact that coalition forces can find IEDs in storage. Instead we chose to combine storage with the construction state so that our found flow rates would adhere to JEIDDO's areas of focus. For future work, one can model the IED supply chain to accommodate for storage. One model for this supply chain is depicted in Figure 5.1. Furthermore, the research in this thesis only focused on IEDs as a whole and only used data from Iraq over a 26 month period. Future work can focus on creating different supply chains for different types of IEDs. Researchers can also decide to perform the same analysis of IED supply chains and focus on different countries, regions or battlefields.

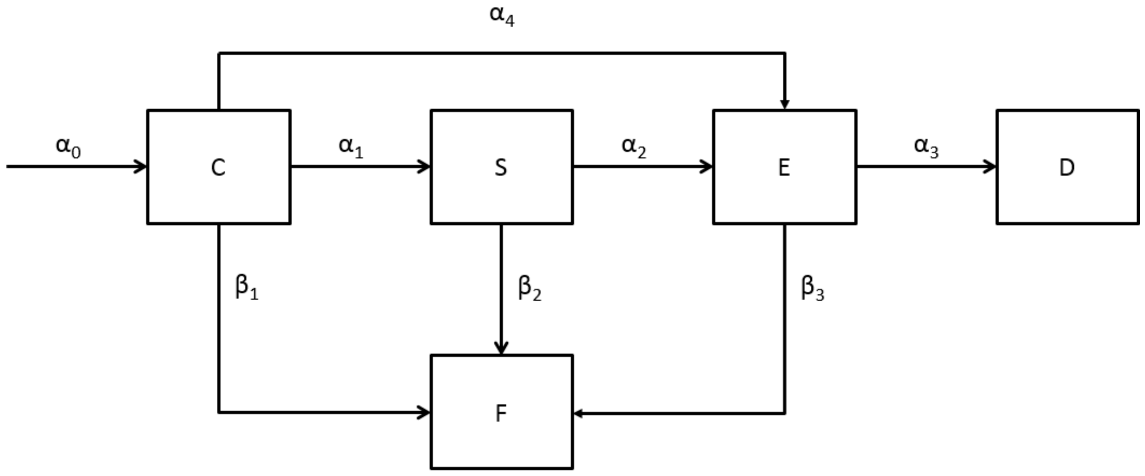


Figure 5.1: IED Supply Chain with Storage

By solving the least squares problem, we were able to look back and study the progression of the construction, emplaced, detonated and found states. In the future, one could use this information to look back and judge the impact of specific coalition C-IED operations. By doing this, we can compare different C-IED operations and determine which operations helped lower the number of IEDs emplaced and in construction and which operations did not. This could provide valuable information on which C-IED operations to fund in the future in the face budget constraints.

In this thesis, we only analyzed one data set. In the future, one could analyze more data sets better modeled with the nonlinear case. One example could be a data set where $\dot{D}(t)$ or $\dot{F}(t)$ fluctuate. We see that this could account for changing coalition and enemy strategies. Furthemore, one could incorporate a system of delayed differential equations into our model to account for the time needed for a change in strategy to take effect.

In the fits for IEDs detonated, we saw that the data points oscillated around our fits. To extend the IED model, one could incorporate oscillatory behavior into the model. The introduction of a oscillating, time dependent behavior, such as seasonal weather effects, can lead us to even better fits and approximations for IED supply chain behavior.

Lastly, one could use the methods provided in this thesis and apply them to combat other threats. For example, drug trafficking supply chains. IED and drug trafficking supply chains have much in common. Traffickers move along drugs throughout the supply chain in a similar manner to enemy forces with IEDs. Furthermore, law enforcement agencies look to intercept drugs in a similar manner to coalition forces with IEDs. The use of parameter estimation methods in this thesis can help law enforcement agencies battle drug trafficking threats.

Bibliography

[1] "Existence Theorems for ODE's". *University of California, Davis*. URL https://www.math.ucdavis.edu/~temple/MAT22C/!!Lectures/3-ExistenceThmsOde-22C-S12.pdf.

[2] "IED Metrics for Iraq: June 2003-September 2010". *Center for Strategic and International Studies*, 11/11 2010. URL https://csis.org/files/publication/101110_ied_metrics_combined.pdf.

[3] "Big Bang Theory: IEDs and Military Countermeasures". *Army-Technology*, 8/19 2011. URL http://www.army-technology.com/features/feature127559/.

[4] "Countering Improvised Explosive Devices". *The White House*, 2/26 2013. URL http://www.whitehouse.gov/sites/default/files/docs/cied_1.pdf.

[5] "IED casualties dropped 50 Percent in Afghanistan in 2012". *USA Today*, 1/18 2013. URL http://www.usatoday.com/story/news/world/2013/01/18/ied-casualties-down-afghanistan-2012/1839609/.

[6] Ackleh, Hearfott, Allen and Seshaiyer. *Classical and Modern Numerical Analysis*. Chapman and Hall, Boca Raton, Florida, 2010.

[7] Arney, David C. and Kristin Arney. "Modeling Insurgency, Counter-Insurgency, and Coalition Strategies and Operations". *The Journal of Defense Modeling and Simulation*, 12/18 2012.

[8] Beamon, Benita M. *Supply Chain Design and Analysis: Models and Methods*. Technical report, University of Washington, 1998.

[9] Briscoe, Erica, Ethan Trewhitt, Lora Weiss, and Elizabeth Whitaker. *Modeling Behavioral Activities Related to Deploying IEDs in Iraq*. Technical report, Georgia Tech Research Institute, 8/28 2009.

[10] Dekker, Anthony H. "Agent-Based Simulation for Counter-IED: A Simulation Science Survey". *Red*, 1:108, 2006.

[11] Dekker, Anthony H. "Optimisation, Games, Adaptation: Three Perspectives on Operations Research for Counter-IED". *Proceedings of SimTecT 2011*, 2011.

[12] Griva, Igor. *Nonlinear Least Squares Data Fitting*. Technical report, George Mason University. URL http://math.gmu.edu/~igriva/book/Appendix%20D.pdf.

[13] JIEDDO. "Counter-Improvised Exposive Device Strategic Plan 2012-2016", 4/25 2011. URL http://www.defenseinnovationmarketplace.mil/resources/20120116_JIEDDO_C-IEDStrategicPlan.pdf.

[14] JIEDDO. “Attack the Network”, 3/7 2013. URL https://www.jieddo.mil/content/docs/20130307_FS_Attack_the_Network.pdf.

[15] JIEDDO. “Defeat the Device”, 3/7 2013. URL https://www.jieddo.mil/content/docs/20130307_FS_Defeat_the_Device.pdf.

[16] JIEDDO. “Train the Force”, 3/7 2013. URL https://www.jieddo.mil/content/docs/20130307_FS_Train_the_Force.pdf.

[17] Mathematica. “”ExplicitRungeKutta” Method for NDSolve”, 2013. URL http://reference.wolfram.com/mathematica/tutorial/NDSolveExplicitRungeKutta.html.

[18] Mathworks. “Choose a Solver”, 2013. URL http://www.mathworks.com/help/gads/choosing-a-solver.html.

[19] Mathworks. “Genetic Algorithm”, 2013. URL http://www.mathworks.com/discovery/genetic-algorithm.html.

[20] Mathworks. “How GlobalSearch and MultiStart Work”, 2013. URL http://www.mathworks.com/help/gads/how-globalsearch-and-multistart-work.html#bsc9ef1-1.

[21] Mathworks. “How the Genetic Algorithm Works”, 2013. URL http://www.mathworks.com/help/gads/how-the-genetic-algorithm-works.html.

[22] Mathworks. “list”, 2013. URL http://www.mathworks.com/help/gads/randomstartpointset.list.html.

[23] Mathworks. “lsqnonlin”, 2013. URL http://www.mathworks.com/help/optim/ug/lsqnonlin.html.

[24] Mathworks. “Minimize Rastrigin’s Function”, 2013. URL http://www.mathworks.com/help/gads/example-rastrigins-function.html.

[25] Mathworks. “ode45”, 2013. URL http://www.mathworks.com/help/matlab/ref/ode45.html.

[26] Mathworks. “RandomStartPointSet class”, 2013. URL http://www.mathworks.com/help/gads/randomstartpointsetclass.html.

[27] Mathworks. “Unconstrained Nonlinear Optimization Algorithms”, 2013. URL http://www.mathworks.com/help/optim/ug/unconstrained-nonlinear-optimization-algorithms.html#f3137.

[28] Price, Jay. “U.S. Handing off Tasks of Tracking IEDs”. *The Seattle Times*, 3/17 2013. URL http://seattletimes.com/html/nationworld/2020569927_afghanbombsxml.html.

[29] Quantock, David E. and Michael A. Vane. “Countering the Improvised Explosive Device Threat”. *Army Magazine*, March 2011. URL http://www.ausa.org/publications/armymagazine/archive/2011/3/Documents/Vane_Quantock_0311.pdf.

[30] Sontag, Eduardo. *Lecture Notes on Mathematical Systems Biology*. Technical report, Rutgers University, 12/3 2013. URL http://www.math.rutgers.edu/~sontag/FTP_DIR/systems_biology_notes.pdf.

[31] Struble, Raimond A. *Nonlinear Differential Equations*. Robert E. Krieger Publishing Company, Malabar, Florida, 1983.

[32] Willy, Christina J. *Robust Sensitivity Analysis for the JEDDO Proposal Selection Model*. Technical report, Air Force Institution of Technology, March 2009.

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704–0188*

1. REPORT DATE *(DD–MM–YYYY)* 27–03–2014	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* Sep 2012–Mar 2014

4. TITLE AND SUBTITLE Modeling Continuous IED Supply Chains	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Liu, Tony, Second Lieutenant, USAF	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB, OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT-ENC-14-M-02
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Colonel Leo Bradley Department of Defense Explosives Safety Board 4800 Mark Center Drive Alexandria, VA 22350-3606 usarmy.pentagon.hqda-dod-esb.mbx.web-team@mail.mil	**10. SPONSOR/MONITOR'S ACRONYM(S)** DDESB
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

Improvised Explosive Devices (IEDs) continue to be a main weapon used by terrorists against coalition forces overseas. This thesis intends to provide methods that can give coalition forces a new perspective on fighting IEDs.
We begin by first developing a model of the supply chain terrorists use to develop, emplace and detonate IEDs. Our model contains four states in which IEDs can exist in: construction (C), emplaced (E), detonated (D) and found by coalition forces (F). We also have rate parameters representing the flow rates of IEDs. Over a given period of time, coalition forces can collect data on the number of IEDs that they find as well as the number of IEDs that detonate. From here, we apply a least squares method to obtain the parameter set for our supply chain model that best fits the collected IED data. Minimizing our least squares equation allows us to estimate where the IEDs are located as well as how fast they are being moved throughout the entire supply chain. Using this, we can judge the impact of our past efforts in stopping IEDs and determine how to best move forward.

15. SUBJECT TERMS

C-IED, Least Squares, Supply Chain, Numerical Differential Equations

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Kevin Pond, PhD (AFIT/ENC)
U	U	U	UU	90	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255-3636 ext.4630, kevin.pond@afit.edu